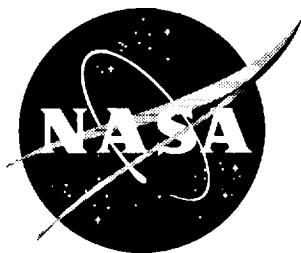


NASA/CR-2000-209863



Motion Cueing Algorithm Development: Initial Investigation and Redesign of the Algorithms

*Robert J. Telban, Weimin Wu, and Frank M. Cardullo
Department of Mechanical Engineering
State University of New York, Binghamton, New York*

March 2000

The NASA STI Program Office ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

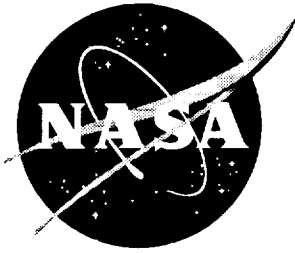
- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.
- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results ... even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at <http://www.sti.nasa.gov>
- E-mail your question via the Internet to help@sti.nasa.gov
- Fax your question to the NASA STI Help Desk at (301) 621-0134
- Phone the NASA STI Help Desk at (301) 621-0390
- Write to:
NASA STI Help Desk
NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076-1320

NASA/CR-2000-209863



Motion Cueing Algorithm Development: Initial Investigation and Redesign of the Algorithms

Robert J. Telban, Weimin Wu, and Frank M. Cardullo
Department of Mechanical Engineering
State University of New York, Binghamton, New York

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Contract NAS1-20454

March 2000

Available from:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 605-6000

Table of Contents

Symbols	3
Abstract	9
1. Introduction	11
2. Background Information	13
2.1. Effect of Simulator Motion and Human Perception Motion	13
2.2. Reference Frames	15
2.3. Coordinate Transformation	18
2.4. Actuator Extensions	21
2.5. Input Scaling and Limiting	22
3. Structure of Simulation Software	25
4. Descriptions of the Four Washout Algorithms	29
4.1. Classical Algorithm	30
4.2. NASA Coordinated Adaptive Washout Algorithm	35
4.3. UTIAS Coordinated Adaptive Washout Algorithm	40
4.4. MIT/UTIAS Optimal Washout Algorithm	45
5. Changes Made in Project to Original Algorithms	57
5.1. Instability in the UTIAS Adaptive Algorithm	59
5.2. Spikes in the Outputs of NASA and UTIAS Adaptive Algorithms ..	63
6. Phase 1 Results	69
6.1. Translational Input Case	69
6.2. Rotational Input Case	73
6.3. Conclusions and Recommendations	75
7. Changes Made in Project for Phase 2	77
7.1. New Vestibular and Semicircular Models	78
7.2. Redefinition of Center of Rotation	79
7.3. Revised Development of Optimal Algorithm	82
7.4. Nonlinear Gain Algorithm	93
7.5. Actuator Extension Limiting	95

8.	Phase 2 Results.....	99
8.1.	Pitch/Surge Mode and Roll/Sway Mode.....	99
8.2.	Heave Mode	101
8.3.	Pitch Mode and Roll Mode	101
8.4.	Yaw Mode	102
8.5.	Braking Algorithm	102
9.	Conclusions	103
	References	105
	Appendix A: Human Semicircular Canal Sensation Model.....	107
	Appendix B Phase 1 Rotational Output Figures	122
	Appendix C: Phase 2 Test Run Output Figures	151

Symbols (unless otherwise noted in report)

\underline{a}	acceleration of a point
$a(t)$	a function of time which increases from 0 to $\pi / 2$ gradually
Δ_j^S	the coordinates of the upper bearing block of the j-th actuator in Fr_S components
B_j^I	the coordinates of the lower bearing block of the j-th actuator in Fr_I components
A, B, C, D	matrices which define the state-space model of a control system
A'	a matrix in the state-space model of the standard form optimal control system
c_0, c_1, c_2, c_3	coefficients of the nonlinear gain polynomial
$d, e, \gamma, \lambda, \delta, \eta$	NASA adaptive algorithm washout parameters
$E\{ \}$	mathematical mean of a statistical variable
E_1, E_2, E_3, E_4	actuator extensions
\underline{e}	pilot's sensation error
F or $F(t)$	a matrix which relates the states of the optimal control system with the input to the controlled plant
Fr	reference frame
f	specific force = $\underline{a} - \underline{g}$
\hat{f}	sensed specific force
G_0	a constant in the UTIAS otolith sensation model
G_C	UTIAS adaptive algorithm steepest descent parameters
\underline{g}	acceleration due to gravity
H	a matrix related to the white noise input in the state-space model of a control system

\mathbf{I}	identity matrix
J_{sub}	system cost function
K	a constant in the UTIAS otolith sensation model
K_{sub}	NASA adaptive algorithm steepest descent parameters
$k_{\text{sub}}, p_{\text{sub}}$	UTIAS adaptive algorithm washout parameters
ℓ_i	displacement of the i-th hydraulic actuator
\mathbf{L}_{SI}	transformation matrix from frame \mathbf{E}_S into frame \mathbf{E}_I
$\mathbf{P}(\mathbf{t})$	solution of the algebraic Riccati equation
p	roll rate component of $\underline{\omega}$
q	pitch rate component of $\underline{\omega}$
r	yaw rate component of $\underline{\omega}$
$\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \mathbf{R}$	weighing matrices in a cost function
$\mathbf{P}, \tilde{\mathbf{P}}, \mathbf{Q}, \mathbf{R}_d$	weighing matrices in a cost function
\mathbf{R}	radius vector
\underline{S}	simulator centroid displacement
s	Laplace variable
s_0, s_1	slopes of the nonlinear gain polynomial
t	time
\mathbf{T}_S	transformation matrix from angular velocity to Euler angle rates
T_0, T_1, T_2, T_3	coefficients in the semicircular canal sensation model
\mathbf{TF}	transfer function
u, v, w	x-, y-, and z-components of velocity \underline{V}

\underline{u}	input to a control system
\underline{u}'	input to a standard form optimal control system
\underline{v}	filtered white noise
\underline{w}	white noise
\underline{V}	velocity $\underline{V} = [u \ v \ w]^T$
$\mathbf{W}(s)$	optimal control open-loop transfer function matrix
$\underline{x}, \underline{X}$	system state vector
\underline{y}	output of a control system
$\hat{\underline{y}}_1$	output of the pilot's vestibular system
α	rotation angle of the simulator platform
$\underline{\beta}$	Euler angles $\underline{\beta} = [\phi \ \theta \ \varphi]^T$
θ	Euler pitch angle component of $\underline{\beta}$
ϕ	Euler roll angle component of $\underline{\beta}$
ψ	Euler yaw angle component of $\underline{\beta}$
ρ	scalar weight in optimal cost criterion
σ	cost criterion for optimal algorithm
τ_L, τ_a, τ_s	time constants in the semicircular and otolith sensation models
$\underline{\omega}$	angular velocity about the body frame $\underline{\omega} = [p \ q \ r]^T$
ω_b	first-order high-pass filter break frequency
ω_n	second-order system undamped natural frequency
$\hat{\omega}$	sensed angular velocity
ζ	second-order system damping ratio

Subscripts (main symbol)_{subscript}

In most cases, subscripts indicate to what the main symbol is related.

- ()_A relates to aircraft
- ()_{AR} relates to aircraft rotation
- ()_{CA} relates to aircraft point corresponding to simulator centroid
- ()_{CG} relates to center of gravity of aircraft
- ()_d relates to simulator states included in the cost function
- ()_{dis} relates to filtered white noise disturbance model
- ()_{des} relates to desired output
- ()_I relates to inertial reference frame
- ()_L relates to low pass filtering
- ()_n relates to white noise input states
- ()_{oto} relates to otolith model
- ()_{PA} relates to pilot in aircraft
- ()_{PS} relates to pilot in the simulator
- ()_r relates to reference model
- ()_S relates to simulator
- ()_{SR} relates to simulator rotation
- ()_{ssc} relates to semicircular canals sensation model
- ()_{ST} relates to simulator tilt coordination
- ()_V relates to pilot's vestibular model
- ()_{x,y,z} x,y, or z components

Superscripts (main symbol)^{superscript}

In most cases, superscripts indicate which frame the main symbol is in

$()^A$ in aircraft frame Fr_A

$()^I$ in inertial frame Fr_I

$()^S$ in simulator frame Fr_S

This Page Intentionally Left Blank.

Abstract

This project was conducted in two phases. In Phase 1, four algorithms including the classical algorithm, the NASA adaptive algorithm, the UTIAS adaptive algorithm, and the optimal algorithm were investigated. The classical algorithm generated results with more distortion, more delay and lower magnitude than the results generated by the other three algorithms. The classical algorithm is the fastest one. This is of little importance since today's computers are fast and none of the four algorithms will run beyond the required time. Therefore, the classical algorithm has no advantage and was not considered in Phase 2.

The two adaptive algorithms are basically similar. The NASA algorithm is well tuned with satisfactory performance. The UTIAS adaptive algorithm strives for more flexibility, but results show that it does not behave better than the NASA algorithm while having more undesirable properties. Some changes were made to the adaptive algorithms such as reducing the magnitude of undesirable spikes.

The optimal algorithm was found to have the potential to behave much better than it did in Phase 1. In Phase 2, the optimal algorithm was redesigned. The center of simulator rotation was redefined. More terms were involved in the optimal algorithm cost function to yield more flexibility in tuning the algorithm. A new design approach featuring a Fortran/Matlab/Simulink interactive design was used. Each set of selected parameters could be tested in only 30 seconds while the old design approach could require as much as 15 minutes. This makes it possible to try hundreds of sets of parameters. As a result, the optimal algorithm was well tuned in Phase 2 that also incorporated a revised vestibular sensation model.

The effect of motion was also discussed. The topic covers what type of motion is desirable and what type is undesirable. A new semicircular canal sensation model was constructed and justified. The discussion and the new sensation model helped to develop the optimal algorithm and to evaluate the motion-base driving algorithms.

In Phase 2 comparisons were made between the NASA adaptive algorithm and the redesigned optimal algorithm. Results showed that the optimal algorithm has some advantages over the NASA adaptive algorithm and might be the best among the four algorithms involved in this study. At the same time the NASA adaptive algorithm was observed to be a very well developed algorithm.

There were some general problems left unresolved in Phase 1 that required solutions. The first problem was that when inputs had large magnitudes, all the algorithms tended to drive the simulator beyond its motion limit. A new nonlinear gain algorithm was designed in Phase 2, the effect of which was that the simulator would not reach its motion limit in most input cases, while the gain in low magnitude input cases remained high. The second problem was that when the simulator reached its limit, an algorithm was needed to brake the simulator to a full stop and then release the brake at some proper time to allow the simulator to follow the output of the washout algorithm again. This braking algorithm was developed in Phase 2.

1. Introduction

With continuing improvement in hardware and software, flight simulation plays an expanding role in the training of aircraft crews, design of new aircraft, and entertainment. This study evaluates a motion-base driving algorithm for a modern six-post synergistic aircraft simulator. Three types of algorithms, classical, adaptive, and optimal, are evaluated within the scope of this study. Some implemented cueing algorithms were first investigated, with some efforts then spent to improve or even re-design them.

The purpose of a motion simulation is to provide task-critical motion and force information (i.e., cues) and any required components of the stressor-induced workload increment that would be present in flight. Since a ground-based flight simulator system cannot duplicate the motions of an actual aircraft, it becomes necessary to determine the best way to utilize its limited capabilities to provide the most necessary and beneficial motion cues. It is also critical for the cueing algorithm to avoid any improper motion cues since it is commonly known that improper motion cues in some flight conditions have great negative effects on the simulation. It is reported that some motion systems experienced being turned off to avoid improper motion cues. A principle component of a motion simulator design is the determination of the motion information that is relevant to the task, can impact human performance, and can be provided within technical and economic constraints. This requires some knowledge regarding the human's motion perception that is therefore also an important portion of this study.

The motion system usually works in conjunction with a visual system to accomplish effective simulation. Although some of the aspects of the visual system are

taken into consideration in this study, a detailed study of the visual system or the co-operation of the two systems is beyond the scope of this research.

This project was conducted in two phases. In Phase 1, four motion cueing algorithms, the classical algorithm, the NASA adaptive algorithm, the UTIAS adaptive algorithm, and the optimal algorithm were investigated. The results of these algorithms were compared with modifications made to correct problems observed with both the NASA adaptive algorithm and the UTIAS adaptive algorithm. In Phase 2, the NASA adaptive algorithm and a redesigned optimal algorithm were further investigated, with the optimal algorithm incorporating a revised model of the human vestibular system. General problems common to all algorithms not resolved in Phase 1 were also addressed in Phase 2.

The motion cueing algorithms are intended to drive the MCFADDEN 676B-B046 simulator at the NASA Langley Research Center. The simulator is a hydraulic six-post synergistic full motion simulator.

2. Background Information

2.1. Effect of Simulator Motion and Human Motion Perception

The goal of the motion system, along with the visual system, is to provide a virtual environment to the pilot so that in a simulator the pilot can perform the controls and maneuvers which are consistent with how they are to be performed in a real aircraft. The simulation can be employed to help in the design of new aircraft, the training of pilots, and research. What is critical to understand is whether motion helps the simulation and what kind of motion is really desirable. Although early studies showed some controversy on the effect of motion on aircraft simulation, results of further studies converged to some consistent conclusions.

2.1.1. Effect of Motion Cue

Gundry [1] reported that Douvillier, et al., Matheny, et al., Perry and Naish, and Tremblay, et al. found that although motion did not always help to reduce pilot performance error in some simulation tests, there were always differences in the pilot control activity power spectra. When motion was provided, there was an increase in the occurrence of high-frequency/low amplitude control movements. These changes served to make the control activity in the moving simulator appear more like that observed during flight than that recorded in a fixed base simulator. The presence of motion was found to reduce phase lag, increase the mid-frequency gain and crossover frequency, and reduce the size of the remnant.

Gundry [1] also reported that Sadoff, et al., and Meiry found that when the simulator dynamics were stable, the presence of maneuver motion did little to improve control. But as the vehicle became unstable, maneuver motion became more important;

its presence allowed the operator to exercise control even in regions where control by visual cues alone would be impossible.

The visual system alone could provide motion illusions in many simulation tests. It was found that as long as there was some proper simulator motion at the beginning of each maneuver, the motion illusions introduced by the visual system could be established more easily and faster. In some simulation cases the motion illusions could exist all the time after its establishment without continuing involvement of simulation motion. This implies that the simulation motion onset requires high attention.

Improper simulator motion could be very harmful for it might conflict with the visual cue and then break the motion illusions introduced by the visual system. The motion cue that conflicts with the visual cue is called a negative motion cue in the literature. Negative motion cues should always be avoided whenever possible.

Clark, et al. [2] found that a pilot's vestibular system could process low level, constant acceleration in the presence of vibratory acceleration as efficiently as it could without the vibratory noise. In other words, vibratory acceleration had little or no masking effect on the detection of constant acceleration over a wide range of intensity levels of constant acceleration. This implies that it is not practical to employ vibrations to mask some motion sensation such as the rotational sensation when the simulator tilts to simulate sustained linear acceleration.

2.1.2. Human Motion Perception

As pointed out in the UTIAS report [3], [4], deriving the human's motion sensation models is important for the evaluation of motion-base drive algorithms and the formulation of the optimal control algorithm. The human vestibular system located in the

head is found to be dominant in human motion sensation. The vestibular system consists of two important parts. One part is the semicircular canals that sense rotational motion and the other are the otoliths that sense linear motion. The UTIAS report [3], [4] provided models for both the semicircular canals and the otoliths.

The semicircular canal sensation model is given as:

$$\frac{\hat{\underline{\omega}}}{\underline{\omega}} = \frac{\tau_L \tau_a s^2}{(\tau_L s + 1)(\tau_s s + 1)(\tau_a s + 1)} \quad (2-1-1)$$

where $\underline{\omega}$ is the angular velocity input and $\hat{\underline{\omega}}$ is the sensed angular velocity. τ_L , τ_s , and τ_a are time constants. The model represents a second order torsion pendulum with an adaptation term. τ_L has a unique value for each rotational degree of freedom: 6.1 sec (roll input p), 5.3 sec (pitch input q), and 10.2 sec (yaw input r). $\tau_s = 0.1$ sec and $\tau_a = 30$ sec for all three degrees of freedom.

The otolith sensation model is given as:

$$\frac{\hat{\underline{f}}}{\underline{f}} = \frac{K(\tau_a s + 1)}{(\tau_L s + 1)(\tau_s s + 1)} \quad (2-1-2)$$

where \underline{f} is the input specific force and $\hat{\underline{f}}$ is the sensed specific force. τ_L , τ_s , and τ_a are time constants and K is also a constant. $\tau_L = 5.33$ sec, $\tau_s = 0.6$ sec, $\tau_a = 13.2$ sec, $K = 0.4$.

2.2.Reference Frames

In describing the development of motion cueing algorithms it is convenient to employ several reference frames. These reference frames are defined below and are shown in Figure 2.1.

2.2.1. Frame Fr_S

The simulator reference frame Fr_S has its origin at the centroid of the simulator payload platform, i.e. the centroid of the simulator's upper bearing attachment points. It is fixed with respect to the simulator payload platform. X_S points forward and Z_S downward with respect to the simulator cockpit. Y_S points toward the pilot's right hand side. The x-y plane is parallel to the floor of the cockpit.

2.2.2. Frame Fr_A

The aircraft reference frame Fr_A has its origin at the same relative cockpit location as the simulator reference frame Fr_S . Fr_A has the same orientation for X_A , Y_A , and Z_A with respect to the cockpit as the simulator frame Fr_S .

2.2.3. Frame Fr_{CG}

The aircraft center of gravity reference frame Fr_{CG} has its origin at the center of gravity of the aircraft. Frame Fr_{CG} has an orientation for X_{CG} , Y_{CG} , and Z_{CG} that is parallel to reference frames Fr_S and Fr_A .

2.2.4. Frame Fr_{PS}

This is a frame attached to the simulator pilot's head with its origin located at a point midway between his left and right vestibular systems. Frame Fr_{PS} has an orientation for X_{PS} , Y_{PS} , and Z_{PS} which is parallel to Fr_A and Fr_S .

2.2.5. Frame Fr_{PA}

This is a frame attached to the aircraft pilot's head with its origin located at a point midway between his left and right vestibular systems. Frame Fr_{PA} has the same orientation for X_{PA} , Y_{PA} , and Z_{PA} as Fr_{PS} .

2.2.6. Frame Fr_I

The inertial reference frame Fr_I is earth-fixed with Z_I aligned with the gravity vector \underline{g} . Its origin is located at the center of the fixed platform motion base. X_I points forward and Y_I points to the right hand side with respect to the simulator pilot.

2.2.7. Reference Frame Locations

In Figure 2.1 are four vectors which define the relative location of the reference frames. R_I defines the location of Fr_S with respect to Fr_I . R_S defines the location of Fr_{PS} with respect to Fr_S . Similarly, R_A defines the location of Fr_{PA} with respect to Fr_A . R_{CG} defines the location of Fr_A with respect to Fr_{CG} .

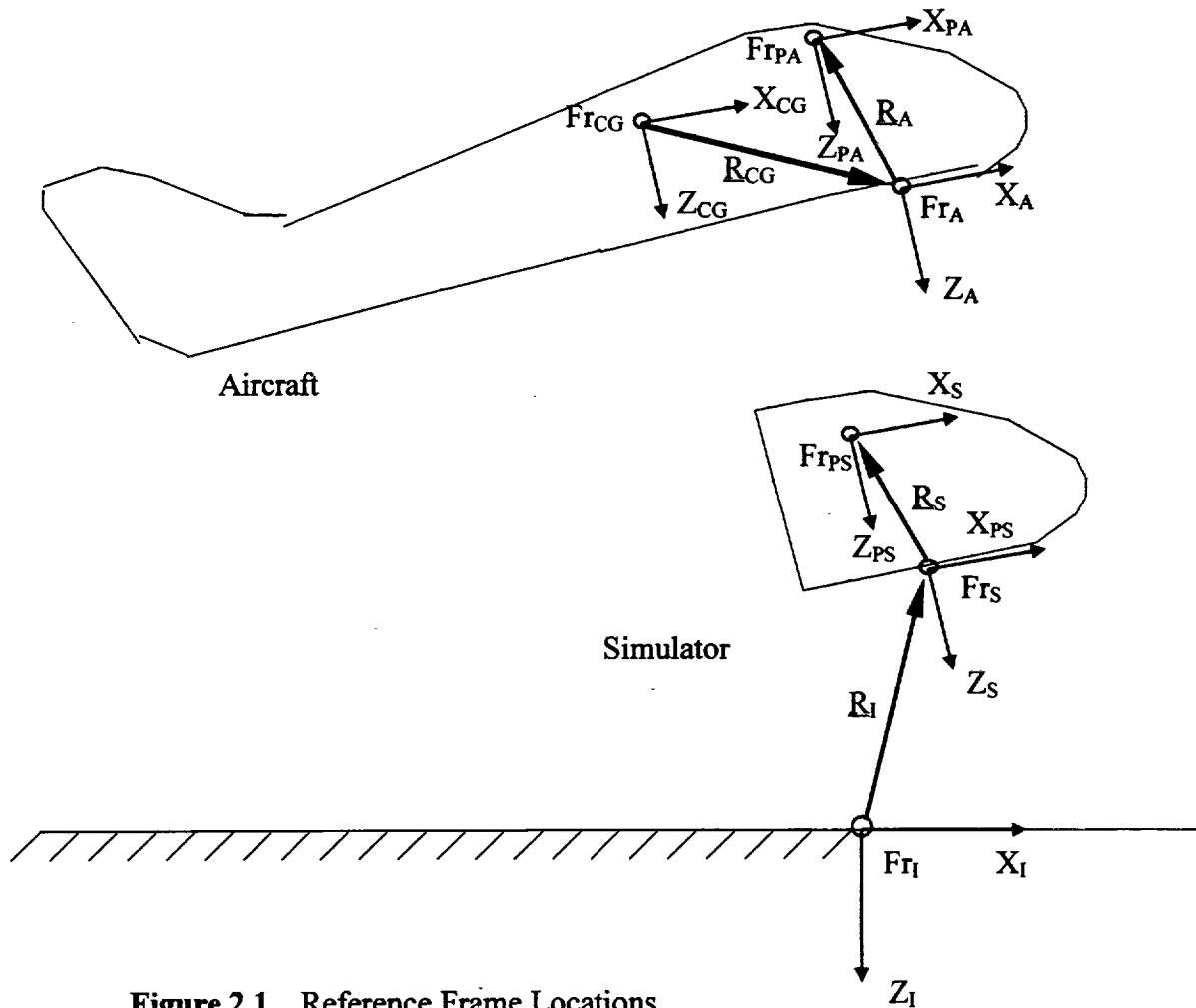


Figure 2.1. Reference Frame Locations.

2.3. Coordinate Transformation

Consider a body having both translational and rotational motion with respect to Fr_I . In Figure 2.2, the inertial translational acceleration of a point b on the body located at a distance $\underline{\delta}_b^B = \delta_x \hat{i} + \delta_y \hat{j} + \delta_z \hat{k}$ from the origin of any body reference frame Fr_B can be expressed by:

$$\underline{a}_b^I = \underline{a}_B^I + \ddot{\underline{\delta}}_b^B + 2\underline{\omega}_B^I \times \dot{\underline{\delta}}_b^B + \dot{\underline{\omega}}_B^I \times \underline{\delta}_b^B + \underline{\omega}_B^I \times (\underline{\omega}_B^I \times \underline{\delta}_b^B) \quad (2-3-1)$$

where

\underline{a}_B^I is the acceleration of the origin of Fr_B with respect to Fr_I .

$\underline{\omega}_B^I$ is the angular rate of Fr_B with respect to Fr_I .

p, q, r are the three components of $\underline{\omega}_B^I$, i.e., $\underline{\omega}_B^I = p \hat{i} + q \hat{j} + r \hat{k}$.

Assuming point b is fixed with respect to Fr_B , then $\ddot{\underline{\delta}}_b^B = \dot{\underline{\delta}}_b^B = 0$.

$$\begin{aligned} \underline{a}_b^I &= \underline{a}_B^I + \dot{\underline{\omega}}_B^I \times \underline{\delta}_b^B + \underline{\omega}_B^I \times (\underline{\omega}_B^I \times \underline{\delta}_b^B) \\ &= \underline{a}_B^I + [-\delta_x(q^2 + r^2) + \delta_y(pq - \dot{r}) + \delta_z(pr + \dot{q})] \hat{i} \\ &\quad + [\delta_x(pq + \dot{r}) - \delta_y(p^2 + r^2) + \delta_z(qr - \dot{p})] \hat{j} \\ &\quad + [\delta_x(pr - \dot{q}) + \delta_y(qr + \dot{p}) - \delta_z(p^2 + q^2)] \hat{k} \end{aligned} \quad (2-3-2)$$

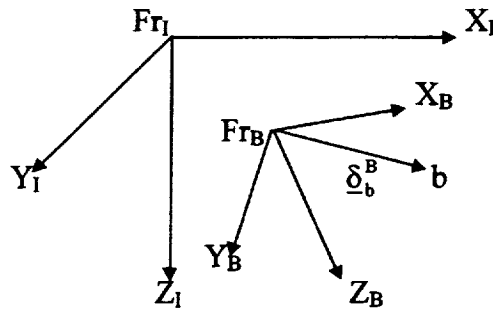


Figure 2.2. Coordinate System for Inertial Acceleration.

2.3.1. Euler Angles

The orientation between two reference frames Fr_I and Fr_B can be specified by three Euler angles: $\underline{\beta} = [\phi \ \theta \ \psi]^T$ which define a sequence of rotations which will carry Fr_I into Fr_B . A vector \underline{V} expressed in the two frames can be related by the transformation matrix $\underline{L}_{IB} (Fr_I \rightarrow Fr_B)$ or $\underline{L}_{BI} (Fr_B \rightarrow Fr_I)$:

$$\underline{V}^B = \underline{L}_{IB} \underline{V}^I \text{ and } \underline{V}^I = \underline{L}_{BI} \underline{V}^B$$

where

$$\underline{L}_{IB} = \underline{L}_{BI}^{-1} = \underline{L}_{BI}^T \quad (2-3-3)$$

$$\underline{L}_{BI} = \begin{bmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix}$$

The angular velocity of Fr_B with respect to Fr_I can be related to the Euler angle rates $\dot{\underline{\beta}}$ by the following. Let $\underline{\omega}_B^I$ represent the components of this angular velocity in frame Fr_B , then

$$\dot{\underline{\beta}} = \underline{T}_B \underline{\omega}_B^I$$

where

$$\underline{T}_B = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \quad (2-3-4)$$

and

$$\underline{\omega}_B^I = \underline{T}_B^{-1} \dot{\underline{\beta}}$$

where

$$\underline{T}_B^{-1} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix}$$

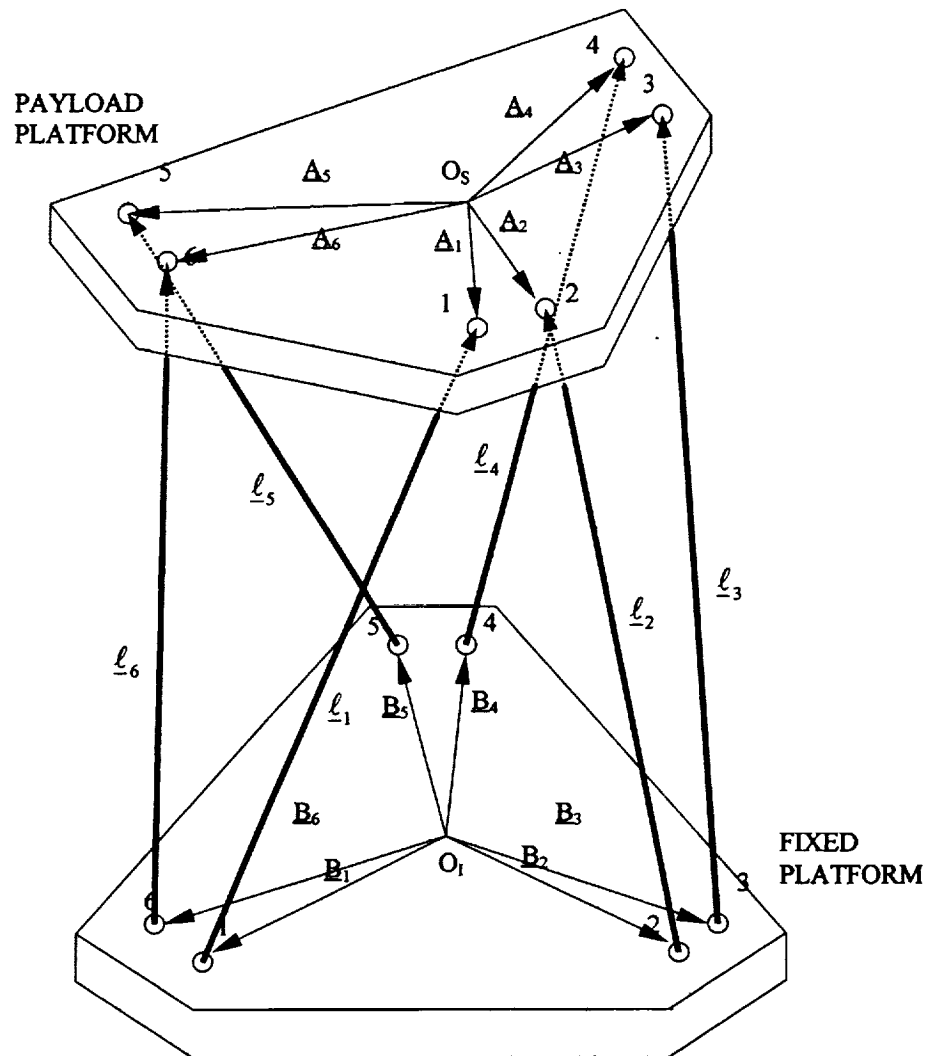


Figure 2.3. Illustration of a Six Post Synergistic Motion Base Geometry.

2.4. Actuator Extensions

The geometry of a six degree-of-freedom synergistic motion base is given in Figure 2.3. The relevant vectors relating the locations of the upper and lower bearings of the j -th actuator are shown in Figure 2.4:

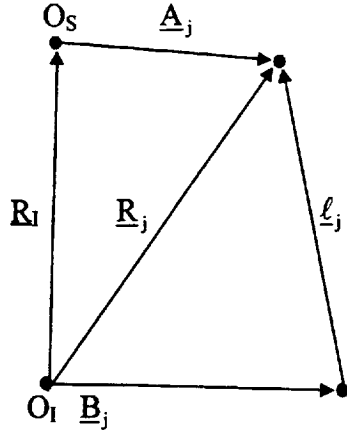


Figure 2.4. Vectors for the j -th actuator.

In Figure 2.4 O_s and O_l are the centers of the payload platform and fixed platform respectively. O_s and O_l are respectively the origins for Fr_s and Fr_l .

It can be seen that the relation among those vectors is:

$$\underline{R}_l + \underline{A}_j^l = \underline{R}_j = \underline{B}_j^l + \underline{l}_j \quad (2-4-1)$$

Then the actuator length vector can be found from:

$$\underline{l}_j = \underline{A}_j^l + \underline{R}_l - \underline{B}_j^l \quad (2-4-2)$$

Expression of \underline{l}_j in Fr_l is desired:

$$\begin{aligned} \underline{l}_j^l &= \underline{A}_j^l + \underline{R}_l - \underline{B}_j^l \\ &= \underline{L}_{st} \underline{A}_j^s + \underline{R}_l - \underline{B}_j^l \end{aligned} \quad (2-4-3)$$

Where \underline{A}_j^s are the coordinates of the upper bearing attachment point of the j-th actuator in Fr_s and \underline{B}_j^l are the coordinates of the lower bearing attachment point of the j-th actuator in Fr_l .

The actuator extensions can then be found from

$$\begin{aligned}\Delta \underline{\ell}_j^l(t) &= \underline{\ell}_j^l(t) - \underline{\ell}_j^l(0) \\ &= (\underline{L}_{sl}(t) - \underline{L}_{sl}(0))\underline{A}_j^s + (\underline{S}^l(t) - \underline{S}^l(0))\end{aligned}\quad (2-4-4)$$

Usually $\underline{L}_{sl}(0) = \underline{I}$ and $\underline{S}^l(0) = 0$, where \underline{I} is the identity matrix. Therefore

$$\Delta \underline{\ell}_j^l = \Delta \underline{L}_{sl} \underline{A}_j^s + \Delta \underline{S}^l \quad (2-4-5)$$

Using small angle approximation $\Delta \underline{L}_{sl}$ can be expressed as:

$$\Delta \underline{L}_{sl} = \begin{bmatrix} 0 & -\psi_s & \theta_s \\ \psi_s & 0 & -\phi_s \\ -\theta_s & \phi_s & 0 \end{bmatrix} \quad (2-4-6)$$

Employing Equations (2-4-5) and (2-4-6) the actuator extensions can then be calculated.

It is observed that a smaller \underline{A}_j^l will result in smaller actuator extensions for a given simulator rotation angle. This information might be useful for simulator hardware design.

2.5. Input Scaling and Limiting

Scaling and limiting are applied to both aircraft translational input signals \underline{a}_A^A and rotational input signals $\underline{\omega}_A^A$. Scaling and limiting modify the amplitude of input uniformly across all frequencies. Limiting is a nonlinear process that clips the signal so

that it is limited to be less than a preselected magnitude. Scaling and limiting can be used to reduce the motion response of a flight simulator. Two input scaling and limiting algorithms were used in the current simulation software. They were suggested and used in [3], [4], and [5].

2.5.1. Linear Input Scaling in Combination with an Input Limiting

The first algorithm is characterized by a linear input scaling in combination with an input limiting. Each component of \underline{a}_A^{\wedge} and $\underline{\omega}_A^{\wedge}$ in different degrees of freedom is scaled and limited separately but in the same manner. The scaling and limiting of a_{Ax} , the x component of \underline{a}_A^{\wedge} is given as an example, where S_x is the slope from $-X_1$ to X_1 :

$$a_x = \begin{cases} S_x a_{Ax} & |a_{Ax0}| \leq X_1 \\ S_x X_1 & a_{Ax} \geq X_1 \\ -S_x X_1 & a_{Ax} \leq -X_1 \end{cases} \quad (2-5-1)$$

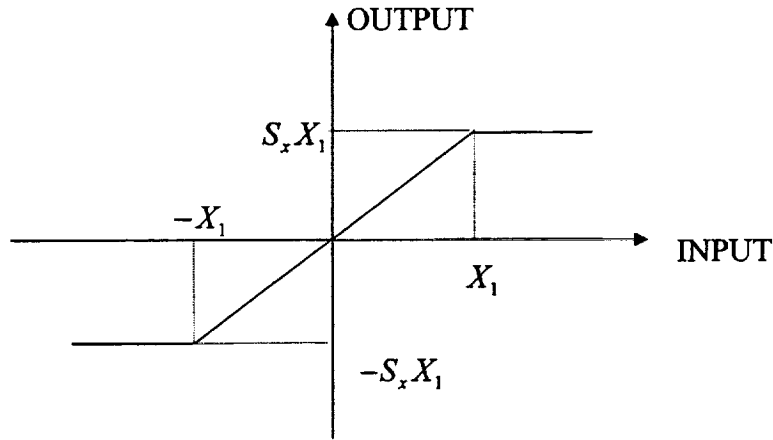


Figure 2.5.1. Linear Input Scaling.

2.5.2. Nonlinear Input Scaling

The second algorithm is characterized by a nonlinear input scaling. Input limiting is not used. Each component of \underline{a}_A^{\wedge} and $\underline{\omega}_A^{\wedge}$ in different degrees of freedom is scaled

separately but in the same manner. The scaling of a_{Ax} , which is the x component of \underline{a}_A , is used as an example:

$$a_x = \begin{cases} S_x a_{x0} & |a_{x0}| \leq X_1 \\ S_x a_{x0} - 0.7S_x(a_{x0} - X_1) & a_{x0} > X_1 \\ S_x a_{x0} - 0.7S_x(a_{x0} + X_1) & a_{x0} < -X_1 \end{cases} \quad (2-5-2)$$

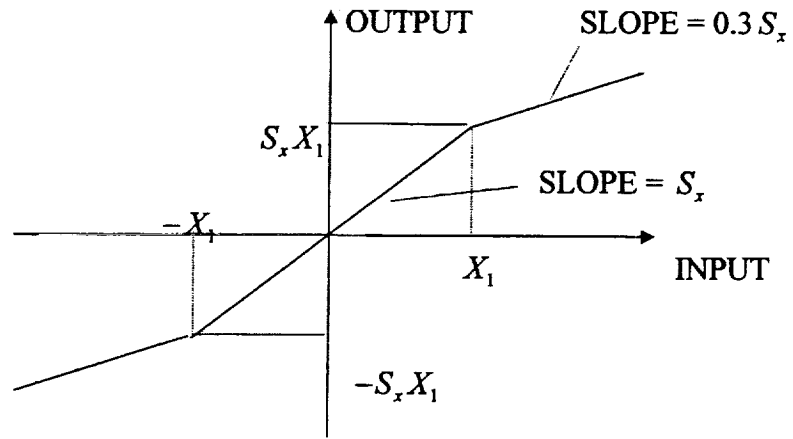


Figure 2.5.2. Nonlinear Input Scaling.

3. Structure of Simulation Software

The execution of the software begins with input of a set of simulation commands. The simulation commands contain selection of input degree of freedom, input type, input magnitude and duration, duration of simulation, and type of cueing algorithm. Each of the input channels for six degrees of freedom can be selected. Maximum duration of simulation is set to be 40 seconds. There are six aircraft input options: step, pulse, pulse doublet, sinusoidal, half sine pulse, and ramp to step. Aircraft translational acceleration at the point corresponding to the centroid of the simulator payload platform \underline{a}_{CA}^A and aircraft angular acceleration $\underline{\omega}_A^A$ are used as input vectors.

From the aircraft inputs the aircraft response and the simulator response are both calculated. Next the aircraft pilot's sensation and the simulator pilot's sensation are calculated and compared. Actuator extensions are generated based on the simulator responses output by the cueing algorithm.

The aircraft response is assumed to follow the input command without error. Then the acceleration at the pilot's head \underline{a}_{PA}^A can be calculated from \underline{a}_{CA}^A and $\underline{\omega}_A^A$. By subtracting the gravity vector \underline{g} the specific force on the pilot's head \underline{f}_A^A can next be calculated. By passing $\underline{\omega}_A^A$ and \underline{f}_A^A through the vestibular model the aircraft pilot's sensation will be generated.

The simulator response is calculated by passing the aircraft commands through the selected cueing algorithm and the platform dynamics filter. The cueing algorithms are the kernels of this software. They are responsible for maximizing the motion cueing effects while restricting the physical motion to be within the displacement, velocity and acceleration capacity of the motion system hardware. The cueing algorithm outputs the

desired translational and rotational platform positions \underline{S}_{des}^I and $\underline{\beta}_{Sdes}$ which are used to compose the desired actuator commands $\underline{\ell}_{des}^I$ that will drive the simulator platform.

In a real simulation, the platform dynamics will cause some error between the platform motion and the motion commands. Passing the motion commands through a filter that is a model of the platform dynamics can simulate this. Then the filter outputs $\underline{\beta}_s$ and \underline{S}^I are assumed to be the real platform positions. Based on $\underline{\beta}_s$ and \underline{S}^I both the simulator translational acceleration \underline{a}_s^I and angular rate $\underline{\omega}_s^S$ may be obtained. The specific force on the simulator pilot's head \underline{f}_{ps}^S is also available. By passing $\underline{\omega}_s^S$ and \underline{f}_{ps}^S through the vestibular model both the sensed angular rate and sensed specific force are obtained. The flowchart of the simulation software is shown in Figure 3.1.

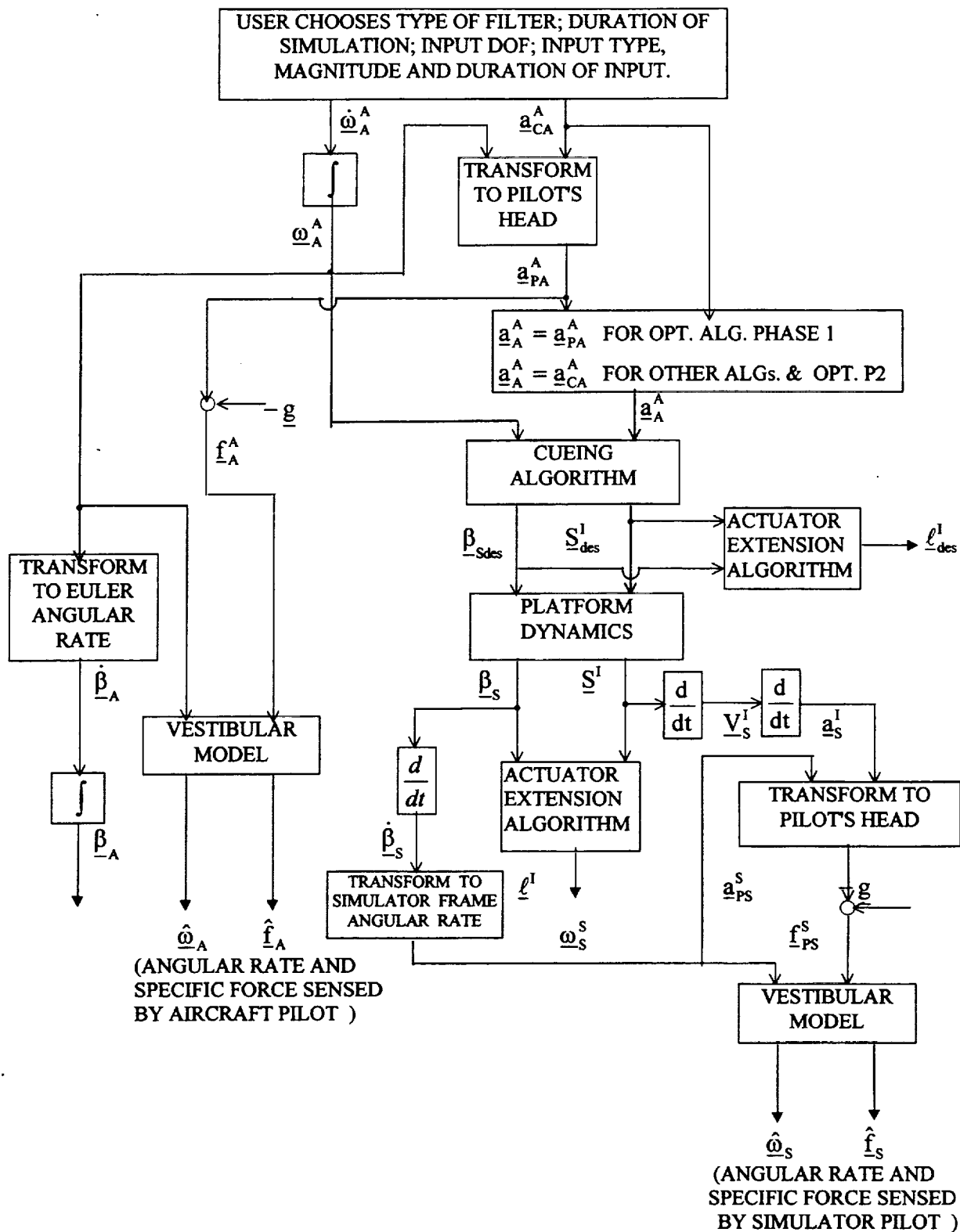


Figure 3.1. Overall Structure of the Software

This Page Intentionally Left Blank.

4. Descriptions of the Four Washout Algorithms

Four algorithms are used in this study. The first is known as the classical algorithm. This type of algorithm is generally denoted in the literature as a linear cueing algorithm [3], [4]. The second algorithm evaluated in this study is the NASA Langley Research Center developed "Coordinated Adaptive Washout Algorithm" [5]. The next cueing algorithm reviewed in the study is a variation of the Coordinated Adaptive Washout Algorithm. This was developed by L. Reid and M. Nahon at the University of Toronto [3], [4]. The Optimal algorithm, the fourth algorithm employed in this study is that developed at MIT by Sivan, et al. [6] and implemented as described in [3] and [4].

The basic task of the washout algorithms is to create a specific force vector and an angular velocity vector at the pilot's location in the simulator approximating those that the pilot would experience in an actual aircraft. The translational and rotational motion effects on the simulator pilot are expected to approximate the motion effects on the aircraft pilot:

$$\underline{f}_{PS}^S \approx \underline{f}_{PA}^A, \quad \underline{\omega}_{PS}^S \approx \underline{\omega}_{PA}^A$$

The relation between the specific force acting on the simulator (aircraft) pilot and the specific force at the origin of the simulator (aircraft) frame can be found from Equation

(2-3-2):

$$\begin{aligned} \underline{f}_{PS}^S &= \underline{a}_{PS}^S - \underline{g}^S \\ &= \underline{a}_S^S + \dot{\underline{\omega}}_S^S \times \underline{R}_S + \underline{\omega}_S^S \times (\underline{\omega}_S^S \times \underline{R}_S) - \underline{g}^S \\ &= \underline{f}_S^S + \dot{\underline{\omega}}_S^S \times \underline{R}_S + \underline{\omega}_S^S \times (\underline{\omega}_S^S \times \underline{R}_S) \\ \underline{f}_{PA}^A &= \underline{a}_{PA}^A - \underline{g}^A \\ &= \underline{a}_A^A + \dot{\underline{\omega}}_A^A \times \underline{R}_A + \underline{\omega}_A^A \times (\underline{\omega}_A^A \times \underline{R}_A) - \underline{g}^A \\ &= \underline{f}_A^A + \dot{\underline{\omega}}_A^A \times \underline{R}_A + \underline{\omega}_A^A \times (\underline{\omega}_A^A \times \underline{R}_A) \end{aligned}$$

Usually $\underline{R}_A = \underline{R}_S$, $\underline{\omega}_S^S = \underline{\omega}_{PS}^S$, $\underline{\omega}_A^A = \underline{\omega}_{PA}^A$. Thus the washout algorithms attempt to achieve:

$$\underline{f}_S^S \approx \underline{f}_A^A = \underline{a}_A^A - \underline{g}^A$$

$$\underline{\omega}_S^S \approx \underline{\omega}_A^A$$

\underline{f}_A^A , $\underline{\omega}_A^A$ or \underline{a}_A^A , $\underline{\omega}_A^A$ are used as washout algorithm inputs.

4.1. Classical Algorithm

This algorithm employs aircraft body axes acceleration \underline{a}_A^A and angular velocity $\underline{\omega}_A^A$ as the aircraft state vector elements that provide the input to the cueing algorithm. A linear scaling in combination with a limiting as described in Section 2.5 is applied within the algorithm to modify the input. The architecture of the classical approach is such that there are separate filters for the translational degrees of freedom and the rotational degrees of freedom with a crossover path to provide the steady state or gravity align cues. This algorithm behaves like open-loop control. The details of this algorithm are presented below.

4.1.1. Translational Degrees of Freedom

The aircraft acceleration vector \underline{a}_A^A is first scaled and limited. This scaling can be either linear or nonlinear. It should be noted at this point that it is not the scaling that makes the cueing algorithm either linear or nonlinear, but rather it is the formulation of the washout filters that is responsible for that characteristic. For the classical algorithm a scaling and limiting scheme as described in Section 2.5.1 is used. After scaling and limiting, the aircraft specific force \underline{f}_A^A is computed and then transformed from the

simulator frame Fr_S into the inertial frame Fr_I , whereupon the inertial frame vector subtracts the gravity vector, with the difference \underline{f}_2^I filtered by a high pass filter (transfer function) of the form

$$TF = \frac{s^3}{(s^2 + 2\zeta\omega_n s + \omega_n^2)(s + \omega_b)} \quad (4-1-1)$$

Where ω_n is a second order system undamped natural frequency and ω_b is a first order system break frequency. ζ is a second order damping ratio. The output is then integrated twice to provide the simulator translational position \underline{S}^I .

4.1.2. Rotational Degrees of Freedom

The rotational rate vector is first scaled and limited by the same scheme for the translational channel. The resulting vector is transformed to the Euler angular rate. The Euler angular rate is then filtered through a high-pass filter (transfer function) of the form

$$TF = \frac{s^2}{(s + \omega_b)^2} \quad (4-1-2)$$

The output is then integrated to provide the desired simulator Euler angular position corresponding to the aircraft rotational input.

4.1.3. Tilt Coordination

Sustained translational acceleration is sensed by the pilot as a long term change in the magnitude and direction of the specific force in the absence of rotational motion. This cannot in general be simulated by translational motion due to motion-base travel limits. It is possible to alter the direction of the steady-state specific force experienced by the pilot in the simulator by tilting the cab. It has become common practice in flight simulators to employ cab tilt to simulate the effect of sustained translational inertial

acceleration. Because this tilt coordination process cannot alter the long term magnitude of the specific force vector, it is an approximation to the desired effect. [3], [4]

The formulation of the tilt coordination Euler angles β_{ST} starts with \underline{a}_L^A as shown in Figure 4-1-1. \underline{a}_L^A is formed by filtering \underline{a}_A^A through a low pass filter of the form:

$$TF = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (4-1-3)$$

\underline{f}_L^A can be considered to be the specific force components that are to be simulated through simulator cab tilt. In the absence of other cab rotational displacement and translational motion, \underline{f}_L^A in Fr_A is shown in Figure 4.1.1 (a). By tilting the simulator cab \underline{f}_S^S can be rotated with respect to Fr_S (Figure 4.1.1 (b)). It is found that this change in the direction of the sensed sustained specific force by the simulator pilot is quite useful in the simulation of low frequency aircraft acceleration components. When the desired tilt angle $\alpha_1 = \alpha_0$, $\underline{f}_S^S = k \underline{f}_L^A$, where $k = g / |\underline{f}_L^A|$, \underline{f}_S^S is successfully aligned. It is easy to see that the desired tilt angles α_1 are:

$$\phi_{SL} = \tan^{-1}(-f_{LY}^A / g) \approx -f_{LY}^A / g$$

$$\theta_{SL} = \tan^{-1}(f_{LX}^A / g) \approx f_{LX}^A / g$$

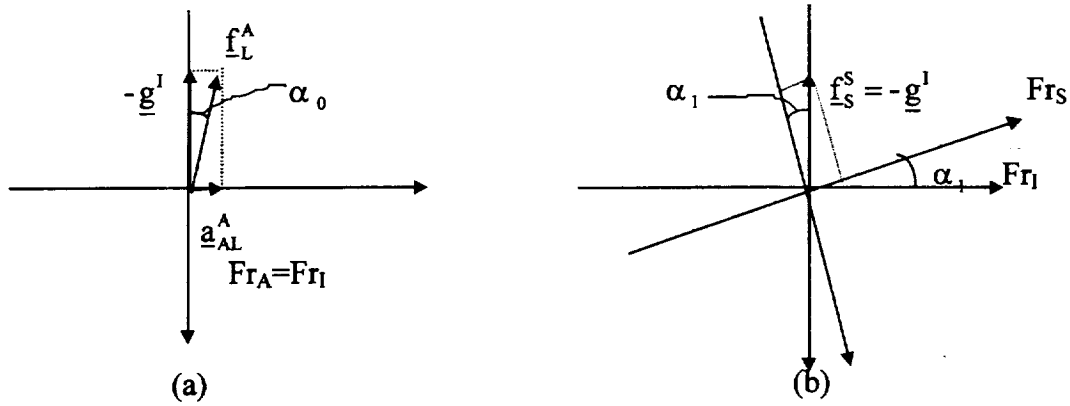


Figure 4.1.1. Simulator Cab Tilt

$\dot{\beta}_{ST}$ is usually restricted under the threshold of pilot perception for avoiding undesired cue.

4.1.4. Summation of Two Rotational Channels

The summation of β_{ST} and β_{SR} will yield β_s , the angular position of the simulator. L_{SI} and T_s can be formed by Equations (2-3-3) and (2-3-4). Then the simulator translational position \underline{S}^I and the angular position β_s can be transformed from degree-of-freedom space to actuator space. These are the actuator lengths required to achieve the desired platform translation.

The block diagram for the classical algorithm is shown in Figure (4-1-2).

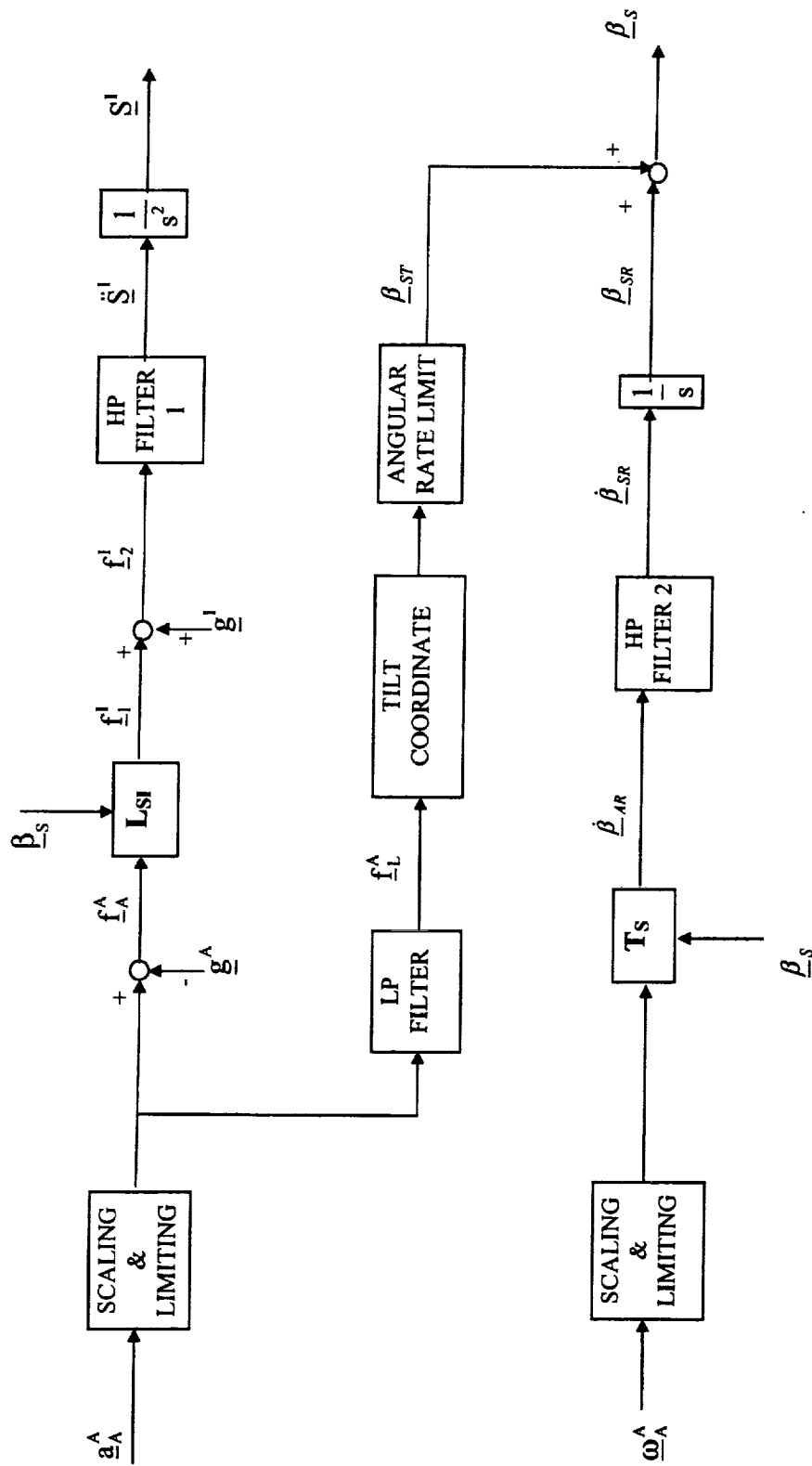


Figure 4.1.2. Classical Algorithm.

4.2. NASA Coordinated Adaptive Washout Algorithm

This algorithm was developed at the NASA Langley Research Center in 1977. The intent of the algorithm is to adapt the severity of the simulator washout filters according to the current state of the simulator. In this way it should be possible to make full use of the simulator motion system at all time. This algorithm also employs \underline{a}_A^{\wedge} and $\underline{\omega}_A^{\wedge}$ as the input to the cueing algorithm. A nonlinear scaling as described in Section 2.5 is applied to modify the input. The general architecture is similar to the classical algorithm in the aspect that there are separate filtering channels for the translational degrees of freedom and the rotational degrees of freedom with a crossover path to provide the steady state or gravity align cues. The block diagram for this type of adaptive algorithm is shown in Figure 4-2. The signals \underline{f}_1^I and $\dot{\underline{\beta}}_A$ represent the inertial frame acceleration components and Euler angles which if applied to the simulator frame Fr_S will produce specific force and rotational velocity components in Fr_S identical to the corresponding Fr_A components in the aircraft. $\dot{\underline{\beta}}_A$ is passed through the rotational channel with an adaptive gain δ to produce a simulator angular rate command. The gravity vector \underline{g} is added to \underline{f}_1^I to yield a new specific force \underline{f}_2^I that will actually be simulated as simulator motion. \underline{f}_2^I is passed through a translational channel with an adaptive gain λ to produce a simulator translational acceleration command and passed through a crossover tilt channel with a fixed gain γ to generate a simulator angular rate command. These commands drive the simulator to the desired translational and angular positions. $\ddot{\underline{S}}^I$ and $\dot{\underline{S}}^I$ are employed as feedback as shown in Figure 4-2. Adaptive parameters λ and δ are continuously adjusted according to the current state of the simulator and

aircraft input. The control equations for this cueing algorithm are presented below. The motion equations are separated and dealt with as four parallel modes: pitch/surge, roll/sway, yaw, and heave.

4.2.1. Pitch/Surge Mode

Control law:

$$\ddot{S}_x^I = \lambda_x \underline{f}_x^I - d_x \dot{S}_x^I - e_x S_x^I \quad (4-2-1)$$

$$\dot{\theta}_s = \gamma_x \underline{f}_x^I + \delta_x \dot{\theta}_A$$

where d_x, e_x & γ_x are fixed parameters for the pitch/surge mode; λ_x and δ_x are the pitch/surge adaptive parameters which are continually adjusted in an attempt to minimize the instantaneous value of the cost function.

Cost function:

$$J_x = \frac{1}{2}(\underline{f}_x^I - \ddot{X})^2 + \frac{W_x}{2}(\dot{\theta}_A - \dot{\theta}_s)^2 + \frac{b_x}{2}X^2 + \frac{C_x}{2}\dot{X}^2 \quad (4-2-2)$$

where W_x, b_x , and C_x are constant weights which penalize the difference in response between the aircraft and simulator, as well as restraining the translational velocity and displacement in the simulator.

Steepest descent for the adaptive parameters:

$$\dot{\lambda}_x = -K_{\lambda_x} \frac{\partial J_x}{\partial \lambda_x} + K_{i\lambda_x} (\lambda_{x0} - \lambda_x) \quad (4-2-3)$$

$$\dot{\delta}_x = -K_{\delta_x} \frac{\partial J_x}{\partial \delta_x} + K_{i\delta_x} (\delta_{x0} - \delta_x)$$

where $K_{\lambda_x}, K_{i\lambda_x}, K_{\delta_x}$, and $K_{i\delta_x}$ are constants. The first right hand side term of each equation in (4-2-3) defines that the change of the adaptive parameter (λ_x or δ_x) is toward a minimum cost function and also defines the rate of change together with the second

right hand side term. The second right hand side term also restrains the deviation of either λ_x or δ_x from their original values.

4.2.2. Roll/Sway Mode

The control law, cost function and steepest descent for the adaptive parameters are in the same form as for the pitch/surge case. Therefore by a substitution of y for x and ϕ for θ the roll/sway motion equations may be obtained. Note that the adaptive parameters, cost function weights and steepest descent constants are unique for the roll/sway mode with a subscript y replacing the subscript x .

4.2.3. Yaw Mode

Control law:

$$\dot{\psi} = \eta_{\psi} \dot{\psi}_A - e_{\psi} \psi_s \quad (4-2-4)$$

where e_{ψ} is a fixed parameter while η_{ψ} is an adaptive parameter.

Cost function:

$$J_{\psi} = \frac{1}{2}(\dot{\psi}_A - \dot{\psi}_s)^2 + \frac{b_{\psi}}{2} \psi_s^2 \quad (4-2-5)$$

where b_{ψ} is a constant weight in the cost function.

Steepest descent:

$$\dot{\eta}_{\psi} = -K_{\eta_{\psi}} \frac{\partial J_{\psi}}{\partial \eta_{\psi}} + K_{in_{\psi}} (\eta_{\psi 0} - \eta_{\psi}) \quad (4-2-6)$$

where $K_{\eta_{\psi}}$ and $K_{in_{\psi}}$ are constants.

4.2.4. Heave Mode

Control law:

$$\ddot{S}_z^I = \eta_z f_z^I - d_z \dot{S}_z^I - e_z S_z^I \quad (4-2-7)$$

where η_z is a fixed parameter while d_z and e_z are adaptive parameters.

Cost function:

$$J_z = \frac{1}{2} (f_z^I - \ddot{Z})^2 + \frac{b_z}{2} Z^2 + \frac{C_z}{2} \dot{Z}^2 \quad (4-2-8)$$

where b_z and C_z are constant weights in the cost function.

Steepest descent:

$$\dot{\eta}_z = -K_{\eta_z} \frac{\partial J_z}{\partial \eta_z} + K_{in_z} (\eta_{z0} - \eta_z) \quad (4-2-9)$$

where K_{η_z} and K_{in_z} are constants.



Figure 4.2. NASA Adaptive Washout Algorithm.

4.3. UTIAS Coordinated Adaptive Algorithm

This algorithm was developed at the University of Toronto in 1985. The general philosophy and structure of this algorithm are the same as those of the NASA adaptive algorithm. The UTIAS adaptive algorithm adapts the severity of the simulator washout filters according to the current state of the simulator. \underline{a}_A^A and $\underline{\omega}_A^A$ are the input to the cueing algorithm. A linear scaling in combination with a limiting as described in Section 2.5 is applied to modify the input. The block diagram is shown in Figure 4-3. The signals \underline{f}_1^I and $\dot{\underline{\beta}}_A$ represent the inertial frame acceleration components and Euler angle rate which if applied to the simulator frame Fr_S will produce specific force and rotational velocity components in Fr_S identical to the corresponding Fr_A components in the aircraft. $\dot{\underline{\beta}}_A$ is passed through the rotational channel with adaptive gain p_{x3} to produce a simulator angular rate command. \underline{f}_2^I is passed through a translational channel with an adaptive gain p_{x1} to produce a simulator translational acceleration command and passed through a crossover tilt channel with an adaptive gain p_{x2} to generate a simulator angular rate command. These commands drive the simulator to the desired translational and angular positions. $\underline{\ddot{S}}^I$ and $\dot{\underline{S}}^I$ are used as feedback as shown in Figure 4-3. Adaptive parameters p_{x1} , p_{x2} and p_{x3} are continuously adjusted according to the current state of the simulator and aircraft input. The difference between the UTIAS and the NASA adaptive algorithms is that the UTIAS algorithm made more parameters adjustable and employed more terms in the cost function in an attempt to improve the effect of the adaptive washout filter. The parameter in the crossover tilt channel is adjustable in the UTIAS algorithm and remains fixed in the NASA algorithm. The absolute values of β and $\dot{\beta}$

along with the deviation of the adaptive parameters from their initial values are employed in the cost function. The control equations for this cueing algorithm are presented below. The motion equations are separated and dealt with as four parallel modes: pitch/surge, roll/sway, yaw, and heave.

4.3.1. Pitch/Surge Mode

Control law:

$$\ddot{S}_x^I = p_{x1} \underline{f}_{IX}^I - k_{x1} S_x^I - k_{x2} \dot{S}_x^I \quad (4-3-1)$$

$$\dot{\theta}_s = \lim (p_{x2} \underline{f}_{IX}^I) + p_{x3} \dot{\theta}_A$$

where k_{x1} and k_{x2} are fixed parameters and parameters p_{x1} , p_{x2} , and p_{x3} are adaptive parameters which are continually adjusted in an attempt to minimize the instantaneous value of the cost function.

Cost function:

$$\begin{aligned} J_x = 0.5 [& \gamma_x (\underline{f}_{IX}^I - \ddot{S}_x^I)^2 + W_{x1} (\dot{\theta}_A - \dot{\theta}_s)^2 \\ & + \rho_x (W_{x2} S_x^{I^2} + W_{x3} \dot{S}_x^{I^2} + W_{x4} \dot{\theta}_s^2 + W_{x5} \theta_s^2) \\ & + W_{x6} (p_{x1} - p_{x10})^2 + W_{x7} (p_{x2} - p_{x20})^2 + W_{x8} (p_{x3} - p_{x30})^2] \end{aligned} \quad (4-3-2)$$

where γ_x , ρ_x , and W_{xi} are constant weights. In (4-3-2) the first group of terms in the cost function penalizes the difference between the aircraft and simulator responses. The second group restrains the translational velocity and displacement of the simulator. The third group restrains the deviation of the adaptive parameters from their original values.

4.3.2. Roll/Sway Mode

The control law and cost function are in the same form as for the pitch/surge case.

Therefore by substituting y for x and ϕ for θ the roll/sway equations may be obtained.

4.3.3. Yaw Mode

Control law:

$$\dot{\psi}_s = p_\psi \dot{\psi}_A - k_{\psi 1} \int \psi_s dt - k_{\psi 2} \psi_s \quad (4-3-3)$$

where $k_{\psi 1}$ and $k_{\psi 2}$ are fixed parameters and p_ψ is an adaptive parameter.

Cost function:

$$J_\psi = 0.5 [(\dot{\psi}_A - \dot{\psi}_s)^2 + \rho_\psi (W_{\psi 1} \dot{\psi}_s^2 + W_{\psi 2} \psi_s^2) + W_{\psi 3} (p_\psi - p_{\psi 0})^2] \quad (4-3-4)$$

where ρ_ψ and $W_{\psi i}$ are constant penalty weights.

4.3.4. Heave Mode

Control law:

$$\ddot{S}_z^i = p_z \underline{f}_{1z}^i - k_{z1} \int S_z^i dt - k_{z2} S_z^i - k_{z3} \dot{S}_z^i \quad (4-3-5)$$

where k_{zi} are fixed parameters and p_z is an adaptive parameter.

Cost function:

$$J_z = 0.5 [(\underline{f}_{1z}^i - \ddot{S}_z^i)^2 + \rho_z (W_{z1} \dot{S}_z^{i2} + W_{z2} S_z^{i2})^2 + W_{z3} (p_z - p_{z0})^2] \quad (4-3-6)$$

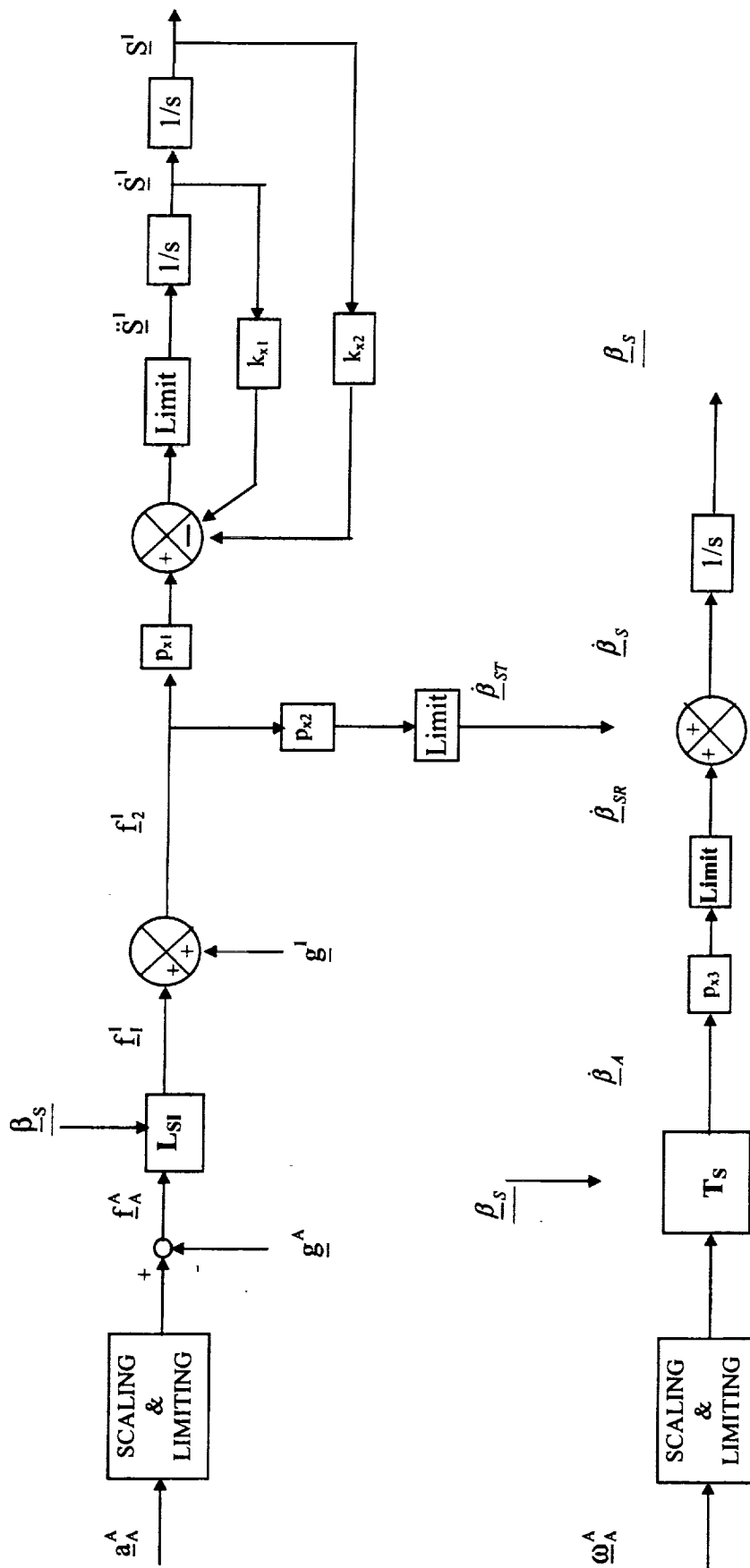
where ρ_z and W_{zi} are constant penalty weights.

4.3.5. Steepest Descent for the Adaptive Parameters

For all four channels the steepest descent has the same form:

$$\dot{p}_{cj} = - G_{cj} \frac{\partial J_c}{\partial p_{cj}} \quad (4-3-7)$$

With the subscript c corresponding to x, y, and z, for ψ the subscript j corresponds to 1, 2, and 3, and G_c are constants.



Note: p_{x1} , p_{x2} , p_{x3} are adaptive parameters and k_{x1}, k_{x2} are constants.

Figure 4.3. UTIAS Adaptive Washout Algorithm.

4.4. MIT/UTIAS Optimal Washout Algorithm

4.4.1. Optimal Control Theory

A linear optimal control theory can be applied to the aircraft simulation problem. The definitions and solutions of linear optimal control problems have some simple standard forms, which make it convenient to apply the optimal control theory and find the solution to a specific optimal control problem.

I. Deterministic Linear Optimal Regulator Problem

The problem can be illustrated by Figure 4.4.1. The problem is to determine some constant matrix F that relates the control system input \underline{u} to the system states \underline{x} and minimizes the optimal criterion J .

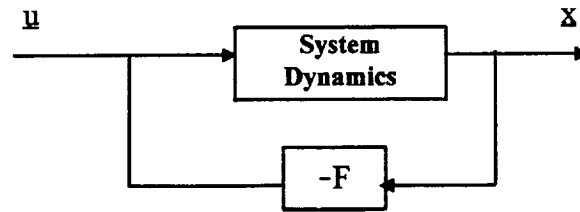


Figure 4-4-1. Deterministic Linear Optimal Regulator.

System equations:

$$\begin{cases} \dot{\underline{x}} = \underline{A}\underline{x} + \underline{B}\underline{u} \\ \underline{y} = \underline{C}\underline{x} \end{cases} \quad \text{with } \underline{x} = \underline{x}(0) \text{ at } t = 0. \quad (4-4-1)$$

$$\text{Criterion: } J = \int_0^{t_1} [\underline{x}^T \underline{R}_1 \underline{x} + \underline{u}^T \underline{R}_2 \underline{u}] dt + \underline{x}^T(t_1) \underline{P}_1 \underline{x}(t_1) \quad (4-4-2)$$

The problem of determining an input \underline{u} for which J is minimized is called the deterministic linear optimal regulator problem.

II. Stochastic Linear Optimal Regulator and Tracking Problems

(a) Regulator Problems with Disturbances: the Stochastic Regulator Problem

The problem can be illustrated by Figure 4.4.2. Disturbance input \underline{v} is a filtered white noise with limited bandwidth. The problem is to determine a constant matrix \mathbf{F} that determines \underline{u} by relating it to the system states \underline{x} and the disturbance input \underline{v} so that the optimal criterion J is minimized.

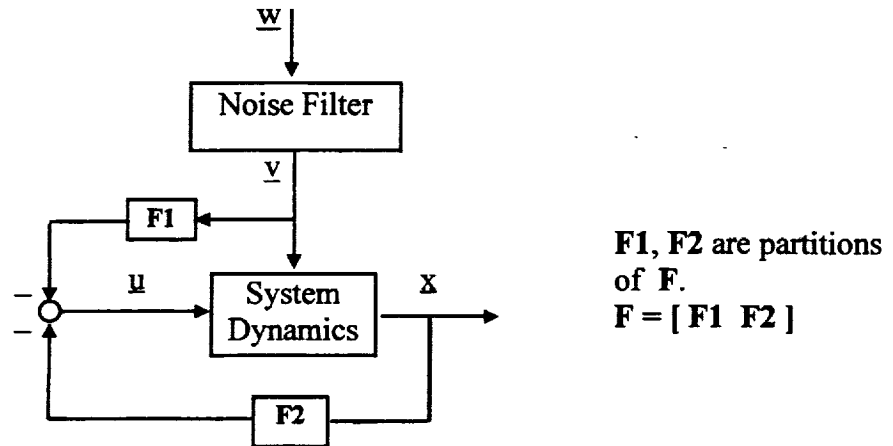


Figure 4.4.2. Stochastic Linear Optimal Regulator.

System equations:

$$\begin{cases} \dot{\underline{x}} = \mathbf{A}\underline{x} + \mathbf{B}\underline{u} + \underline{v} \\ \underline{y} = \mathbf{C}\underline{x} \\ \dot{\underline{x}}_{\text{dis}} = \mathbf{A}_{\text{dis}}\underline{x}_{\text{dis}} + \underline{w} \\ \underline{v} = \mathbf{C}_{\text{dis}}\underline{x}_{\text{dis}} \end{cases} \quad \underline{w} \text{ is white noise.} \quad (4-4-3)$$

Define $\tilde{\underline{x}} = \begin{bmatrix} \underline{x} \\ \underline{x}_{\text{dis}} \end{bmatrix}$, then

$$\begin{cases} \dot{\tilde{\mathbf{x}}} = \begin{bmatrix} \mathbf{A} & \mathbf{C}_{dis} \\ \mathbf{0} & \mathbf{A}_{dis} \end{bmatrix} \tilde{\mathbf{x}} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \underline{\mathbf{u}} + \begin{bmatrix} \mathbf{0} \\ \underline{\mathbf{w}} \end{bmatrix} \\ y = [\mathbf{C} \quad \mathbf{0}] \tilde{\mathbf{x}} \end{cases}$$

Optimal criterion:

$$J = E \left\{ \int_0^T [\underline{\mathbf{y}}^T \mathbf{R}_3 \underline{\mathbf{y}} + \underline{\mathbf{u}}^T \mathbf{R}_2 \underline{\mathbf{u}}] dt + \tilde{\mathbf{x}}^T \mathbf{P}_1 \tilde{\mathbf{x}} \right\} \quad (4-4-4)$$

$$= E \left\{ \int_0^T [\tilde{\mathbf{x}}^T \mathbf{R}_1 \tilde{\mathbf{x}} + \underline{\mathbf{u}}^T \mathbf{R}_2 \underline{\mathbf{u}}] dt + \tilde{\mathbf{x}}^T \tilde{\mathbf{P}}_1 \tilde{\mathbf{x}} \right\} \quad (4-4-5)$$

where $\mathbf{R}_1 = \mathbf{C}^T \mathbf{R}_3 \mathbf{C}$, $\tilde{\mathbf{P}}_1 = \begin{bmatrix} \mathbf{P}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$, \mathbf{R}_1 , \mathbf{R}_2 , \mathbf{R}_3 , and \mathbf{P}_1 are constant matrices, and E is defined as the mathematical mean of a statistical variable.

(b) Stochastic tracking problems.

This problem can be illustrated by Figure 4-4-3. \mathbf{x}_r is the state of a reference system which has a white noise input. \mathbf{x} is the state of the control system which strives for following the reference system. The problem is to determine a constant matrix \mathbf{F} which determines $\underline{\mathbf{u}}$ by relating it to the state of the reference system \mathbf{x}_r and the state of the control system \mathbf{x} so that the optimal criterion J is minimized.

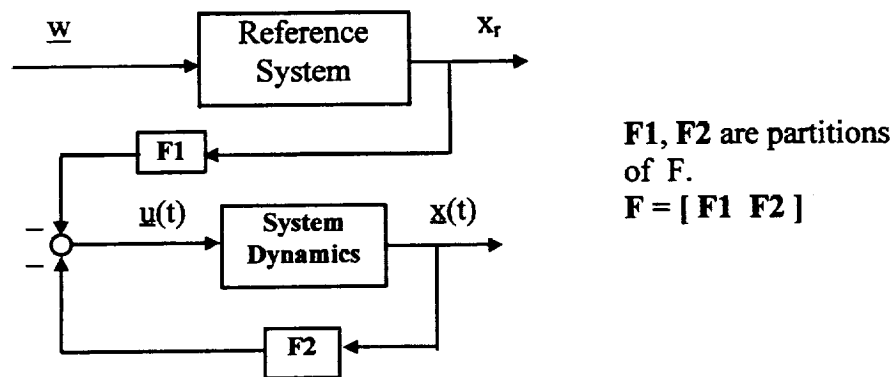


Figure 4.4.3. Stochastic Tracking Problem

System equations:

$$\begin{cases} \dot{\underline{x}} = \underline{A}\underline{x} + \underline{B}\underline{u} \\ \underline{y} = \underline{C}\underline{x} \\ \dot{\underline{x}}_r = \underline{A}_r\underline{x}_r + \underline{w} \\ \underline{y}_r = \underline{C}_r\underline{x}_r \end{cases} \quad \text{w is white noise.} \quad (4-4-6)$$

Define $\tilde{\underline{x}} = \begin{bmatrix} \underline{x} \\ \underline{x}_r \end{bmatrix}$, then

$$\begin{cases} \dot{\tilde{\underline{x}}} = \begin{bmatrix} \underline{A} & \underline{0} \\ \underline{0} & \underline{A}_r \end{bmatrix} \tilde{\underline{x}} + \begin{bmatrix} \underline{B} \\ \underline{0} \end{bmatrix} \underline{u} + \begin{bmatrix} \underline{0} \\ \underline{w} \end{bmatrix} \\ \tilde{\underline{y}} = \underline{y} - \underline{y}_r = [\underline{C} \quad -\underline{C}_r] \tilde{\underline{x}} \end{cases} \quad (4-4-7)$$

Optimal criterion:

$$\begin{aligned} J &= E \left\{ \int_0^1 [(\underline{y} - \underline{y}_r)^T \underline{R}_3 (\underline{y} - \underline{y}_r) + \underline{u}^T \underline{R}_2 \underline{u}] dt \right\} \\ &= E \left\{ \int_0^1 [\tilde{\underline{x}}^T \underline{R}_1 \tilde{\underline{x}} + \underline{u}^T \underline{R}_2 \underline{u}] dt \right\} \\ &\text{where } \underline{R}_1 = [\underline{C} \quad -\underline{C}_r]^T \underline{R}_3 [\underline{C} \quad -\underline{C}_r] \end{aligned} \quad (4-4-8)$$

Equations (4-4-6) and (4-4-8) can be generalized in a common standard form:

$$\text{System equation: } \begin{cases} \dot{\underline{x}} = \underline{A}\underline{x} + \underline{B}\underline{u} + \underline{H}\underline{w} \\ \underline{y} = \underline{C}\underline{x} \end{cases} \quad (4-4-9)$$

$$\text{Optimal criterion: } J = E \left\{ \int_0^1 [\underline{x}^T \underline{R}_1 \underline{x} + \underline{u}^T \underline{R}_2 \underline{u}] dt + \underline{x}^T \underline{P}_1 \underline{x} \right\} \quad (4-4-10)$$

where \underline{R}_1 and \underline{R}_2 are constant matrices and E is defined as the mathematical mean of a statistical variable.

III. Solution to the Optimal Control Problems

There is a common linear optimal solution for the deterministic regulator problem, the stochastic regulator problem and the stochastic tracking problem. The solution has a standard form:

$$\underline{u} = - \mathbf{F} \underline{x} \quad (4-4-11)$$

where

$$\mathbf{F} = \mathbf{R}_2^{-1} \mathbf{B}^T \mathbf{P}(t) \quad (4-4-12)$$

and $\mathbf{P}(t)$ is the solution to the algebraic Riccati equation:

$$-\dot{\mathbf{P}} = \mathbf{R}_1 - \mathbf{P} \mathbf{B} \mathbf{R}_2 \mathbf{B}^T \mathbf{P} + \mathbf{P} \mathbf{A} + \mathbf{A}^T \mathbf{P} \quad (4-4-13)$$

with boundary conditions $\underline{x} = \underline{x}(0)$ at $t = 0$ and $\mathbf{P}(t_1) = \mathbf{P}_1$.

When t_1 approaches infinity, it is proven that \mathbf{P} has a steady-state solution which satisfies the equation:

$$0 = \mathbf{R}_1 - \mathbf{P} \mathbf{B} \mathbf{R}_2 \mathbf{B}^T \mathbf{P} + \mathbf{P} \mathbf{A} + \mathbf{A}^T \mathbf{P} \quad (4-4-14)$$

4.4.2. Aircraft Simulation Problem Definition

The problem is to determine a linear filter matrix $\mathbf{W}(s)$ which relates the simulator motion states \underline{u}_S to the aircraft motion states \underline{u}_A so that some cost function or criterion which constrains the pilot's sensation error \underline{e} and the simulator motion simultaneously will be minimized. The structure of the problem is illustrated in Figure 4.4.4.

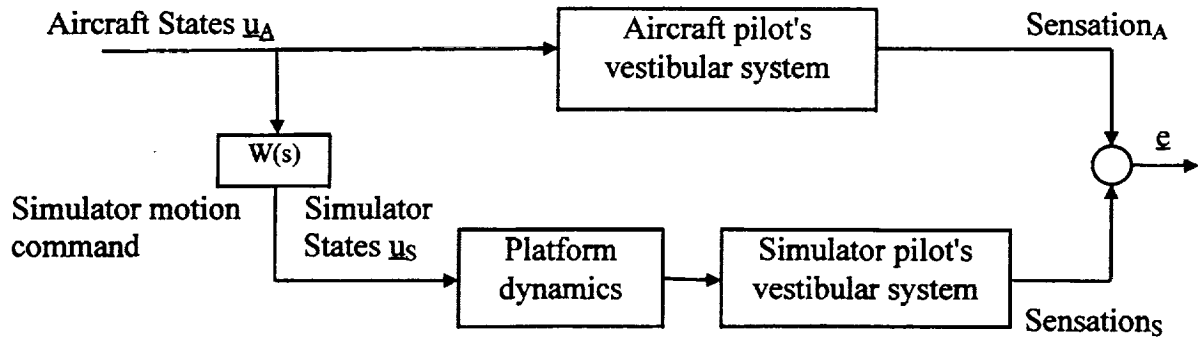


Figure 4.4.4. Aircraft Simulation Problem Structure

Since the control strives for tracking the output of the aircraft pilot's vestibular system, the problem is most likely to be treated as a tracking problem. The aircraft motion can be quite variable. Therefore, it is reasonable to use a filtered white noise that contains sufficient frequency components to represent the aircraft motion states. Therefore, the problem is a stochastic tracking problem.

4.4.3. MIT/UTIAS Development of Washout Filter Coefficients

The optimal algorithm documented in this section was developed at the University of Toronto in 1985. The optimal washout filter design problem is formulated as follows. The actual aircraft and simulator sensation systems $Sensation_A$ and $Sensation_s$ are given along with the input signal u_A that drives $Sensation_A$. A properly constrained operator $W(s)$ can then be found that generates the simulator input u_s to $Sensation_s$ on the basis of the input to $Sensation_A$, such that the pilot sensation error e is as small as possible. A mathematical model of the vestibular organ is used. The optimization criterion that is selected is the mean-square difference between the physiological outputs of the vestibular organs for the pilot in the aircraft and for the pilot

in the simulator. The structure for the optimal filter is assumed given and the task is to find the optimal values for the parameters of the filter.

The optimal algorithm in this section generates the optimized transfer function $W(s)$ by an off-line program. $W(s)$ is then applied on-line to generate simulator motion commands.

$W(s)$ relates the aircraft pilot sensation input to the simulator pilot sensation input that is assumed to be approximately identical to the simulator motion base input

$\underline{u}_s \cong W(s) \underline{u}_A$, where \underline{u}_A consists of aircraft body axes acceleration \underline{a}_A^A and angular displacement $\underline{\beta}_A^A$. \underline{u}_s will be used as the command to drive the motion base. A linear scaling in combination with a limiting as described in Section 2.5 is applied to modify the input.

$W(s)$ is optimized by minimizing the cost function $\sigma = \int_0 J dt$, where

$$J = E \left\{ \underline{e}^T Q \underline{e} + \rho \left(\underline{u}_s^T R \underline{u}_s + \underline{y}_{d_s}^T R_d \underline{y}_{d_s} \right) \right\}$$

$$\underline{e} = \underline{y}_s - \underline{y}_A$$

$$\underline{u}_s^T = [\underline{\beta}_s^T \quad \underline{a}_s^{I^T}]$$

where \underline{y}_s and \underline{y}_a are the pilot's sensations in the aircraft and simulator environments.

For uncoupled system equations, $\underline{y}_{d_s}^T$ has different meanings for different degrees of freedom:

$$\text{For pitch/surge equations: } \underline{y}_{d_s}^T = [\iint \underline{u}_s^I dt^2 \quad \int \underline{u}_s^I dt \quad \underline{u}_s^I]$$

$$\text{For roll/sway equations: } \underline{y}_{d_s}^T = [\iint \underline{v}_s^I dt^2 \quad \int \underline{v}_s^I dt \quad \underline{v}_s^I]$$

$$\text{For yaw equations: } \underline{y}_{d_s}^T = [\iint \underline{\psi}_s dt^2 \quad \int \underline{\psi}_s dt]$$

$$\text{For heave equations: } \underline{y}_{d_s}^T = [\iint \underline{w}_s^I dt^2 \quad \int \underline{w}_s^I dt \quad \underline{w}_s^I]$$

Q and R_d are positive semi-definite matrices, R is a positive definite matrix and ρ is a positive scalar. These parameters determine the weight of each term in the cost function.

The optimal washout is performed in the pilot head frames Fr_{PA} and Fr_{PS} . It is claimed in the UTIAS report [3], [4] that this frame selection will avoid sensation cross-coupling where the sensation cross-coupling may make the calculation of $W(s)$ more complicated.

The washout filter coefficient $W(s)$ will now be generated. The sensation system for the pilot in both the aircraft and simulator environments are given as:

$$\begin{aligned} \text{Sensation}_A: \quad \dot{\underline{x}}_A &= A_1 \underline{x}_A + B_1 \underline{u}_A \\ \underline{\dot{y}}_A &= C_1 \underline{x}_A + D_1 \underline{u}_A \end{aligned} \quad (4-4-15)$$

$$\begin{aligned} \text{Sensation}_S: \quad \dot{\underline{x}}_S &= A_1 \underline{x}_S + B_1 \underline{u}_S \\ \underline{\dot{y}}_S &= C_1 \underline{x}_S + D_1 \underline{u}_S \end{aligned} \quad (4-4-16)$$

Where it has been assumed that the same motion sensing system dynamics can be applied both in the aircraft and in the simulator and all system matrices are taken to be time-invariant.

Assume \underline{u}_A consists of filtered white noise:

$$\begin{aligned} \dot{\underline{x}}_n &= A_n \underline{x}_n + B_n \underline{w} \\ \underline{u}_A &= \underline{x}_n \end{aligned} \quad (4-4-17)$$

Adjoining all the above systems, the optimal controller equations may be generated:

$$\begin{aligned} \dot{\underline{X}} &= A \underline{X} + B \underline{u}_S + H \underline{w} \\ Y &= C \underline{X} + D \underline{u}_S \end{aligned} \quad (4-4-18)$$

with the cost function $\sigma = \int_0^\infty J dt$,

$$\text{where} \quad J = E \{ Y^T G Y + \rho \underline{u}_S^T R \underline{u}_S \} \quad (4-4-19)$$

$$\underline{\mathbf{X}} = \begin{bmatrix} \underline{\mathbf{x}}_A^T \\ \underline{\mathbf{x}}_S^T \\ \underline{\mathbf{x}}_n^T \end{bmatrix} \quad \underline{\mathbf{Y}} = \begin{bmatrix} \underline{\mathbf{e}}^T \\ \underline{\mathbf{y}}_{ds}^T \end{bmatrix} \quad \underline{\mathbf{A}} = \begin{bmatrix} \underline{\mathbf{A}}_1 & \underline{\mathbf{0}} & \underline{\mathbf{B}}_1 \\ \underline{\mathbf{0}} & \underline{\mathbf{A}}_1 & \underline{\mathbf{0}} \\ \underline{\mathbf{0}} & \underline{\mathbf{0}} & \underline{\mathbf{A}}_n \end{bmatrix} \quad \underline{\mathbf{B}} = \begin{bmatrix} \underline{\mathbf{0}} \\ \underline{\mathbf{B}}_1 \\ \underline{\mathbf{0}} \end{bmatrix}$$

$$\underline{\mathbf{C}} = \begin{bmatrix} -\underline{\mathbf{C}}_1 & \underline{\mathbf{C}}_1 & -\underline{\mathbf{D}}_1 \\ \underline{\mathbf{0}} & \underline{\mathbf{C}}_d & \underline{\mathbf{0}} \end{bmatrix} \quad \underline{\mathbf{D}} = \begin{bmatrix} \underline{\mathbf{D}}_1 \\ \underline{\mathbf{0}} \end{bmatrix} \quad \underline{\mathbf{G}} = \begin{bmatrix} \underline{\mathbf{Q}} & \underline{\mathbf{0}} \\ \underline{\mathbf{0}} & \underline{\mathbf{R}}_d \end{bmatrix} \quad \underline{\mathbf{H}} = \begin{bmatrix} \underline{\mathbf{0}} \\ \underline{\mathbf{0}} \\ \underline{\mathbf{B}}_n \end{bmatrix}$$

where ρ and \mathbf{R} are positive scalars.

By matrix substitution and manipulation Equations (4-4-18) and (4-4-19) transform to the standard form of the stochastic linear optimal regulator problem:

$$\dot{\underline{\mathbf{x}}} = \underline{\mathbf{A}}' \underline{\mathbf{x}} + \underline{\mathbf{B}} \underline{\mathbf{u}}' + \underline{\mathbf{H}} \underline{\mathbf{w}} \quad (4-4-20)$$

$$J = E \left\{ \int_0^t [\underline{\mathbf{x}}^T \underline{\mathbf{R}}_1' \underline{\mathbf{x}} + \underline{\mathbf{u}}'^T \underline{\mathbf{R}}_2 \underline{\mathbf{u}}'] dt \right\}$$

whose solution was given in Section 4.4.1:

$$\underline{\mathbf{u}}' = -\underline{\mathbf{F}}_1 \underline{\mathbf{x}},$$

where (4-4-21)

$$\underline{\mathbf{F}}_1 = \underline{\mathbf{R}}_2^{-1} \underline{\mathbf{B}}^T \underline{\mathbf{P}}$$

and $\underline{\mathbf{P}}$ is the solution of the matrix Riccati equation

$$\underline{\mathbf{0}} = \underline{\mathbf{R}}_1' - \underline{\mathbf{P}} \underline{\mathbf{B}} \underline{\mathbf{R}}_2^{-1} \underline{\mathbf{B}}^T \underline{\mathbf{P}} + \underline{\mathbf{A}}'^T \underline{\mathbf{P}} + \underline{\mathbf{P}} \underline{\mathbf{A}}' \quad (4-4-22)$$

with definitions of the new notations used in the above equations:

$$\begin{aligned} \mathbf{R}_1 &= \mathbf{C}^T \mathbf{G} \mathbf{C} ; \mathbf{R}_{12} = \mathbf{C}^T \mathbf{G} \mathbf{D} ; \mathbf{R}_2 = \mathbf{R} + \mathbf{D}^T \mathbf{G} \mathbf{D} ; \\ \mathbf{R}'_1 &= \mathbf{R}_1 - \mathbf{R}_{12} \mathbf{R}_2^{-1} \mathbf{R}_{12}^T ; \underline{\mathbf{u}}' = \underline{\mathbf{u}}_s + \mathbf{R}_2^{-1} \mathbf{R}_{12}^T \underline{\mathbf{x}} ; \\ \mathbf{A}' &= \mathbf{A} - \mathbf{B} \mathbf{R}_2^{-1} \mathbf{R}_{12}^T ; \underline{\mathbf{x}} = \underline{\mathbf{X}} \end{aligned} \quad (4-4-23)$$

Now $\underline{\mathbf{u}}_s$ and $\underline{\mathbf{x}}$ can be related:

$$\underline{\mathbf{u}}_s = \underline{\mathbf{u}}' - \mathbf{R}_2^{-1} \mathbf{R}_{12}^T \underline{\mathbf{x}} = -\mathbf{R}_2^{-1} (\mathbf{B}^T \mathbf{P} + \mathbf{R}_{12}^T) \underline{\mathbf{x}} = -\mathbf{F} \underline{\mathbf{x}} \quad (4-4-24)$$

where $-\mathbf{F} = -\mathbf{R}_2^{-1} (\mathbf{B}^T \mathbf{P} + \mathbf{R}_{12}^T)$.

Partition $-\mathbf{F}$ into $-\mathbf{F}_1 \mathbf{F}_2 \mathbf{F}_3$, then

$$\underline{\mathbf{u}}_s = - \begin{bmatrix} \mathbf{F}_1 & \mathbf{F}_2 & \mathbf{F}_3 \end{bmatrix} \begin{bmatrix} \underline{\mathbf{x}}_A^T \\ \underline{\mathbf{x}}_s^T \\ \underline{\mathbf{x}}_u^T \end{bmatrix} \quad (4-4-25)$$

Taking the LaPlace transform on (4.4.1) and (4.4.3) yields:

$$\begin{aligned} \underline{\mathbf{X}}_s &= (s\mathbf{I} - \mathbf{A})^{-1} \mathbf{B} \underline{\mathbf{u}}_A \\ \underline{\mathbf{X}}_s &= (s\mathbf{I} - \mathbf{A})^{-1} \mathbf{B} \underline{\mathbf{u}}_s \end{aligned} \quad (4-4-26)$$

By the substitution of (4-4-26) into the LaPlace transform of (4.4.25), the relation between $\underline{\mathbf{u}}_s(s)$ and $\underline{\mathbf{u}}_A(s)$ is finally found:

$$\underline{\mathbf{u}}_s(s) = \mathbf{W}(s) \underline{\mathbf{u}}_A(s) \quad (4-4-27)$$

where

$$\mathbf{W}(s) = [-\mathbf{I} + \mathbf{F}_2 (s\mathbf{I} - \mathbf{A} + \mathbf{B} \mathbf{F}_2)^{-1} \mathbf{B}] [\mathbf{F}_1 (s\mathbf{I} - \mathbf{A})^{-1} \mathbf{B} + \mathbf{F}_3]$$

$\mathbf{W}(s)$ is the optimized open-loop transfer function linking $\underline{\mathbf{u}}_s(s)$ and $\underline{\mathbf{u}}_A(s)$, the optimal algorithm controller implemented in the UTIAS report. The block diagram for the optimal algorithm is shown in Figure 4.4.5.

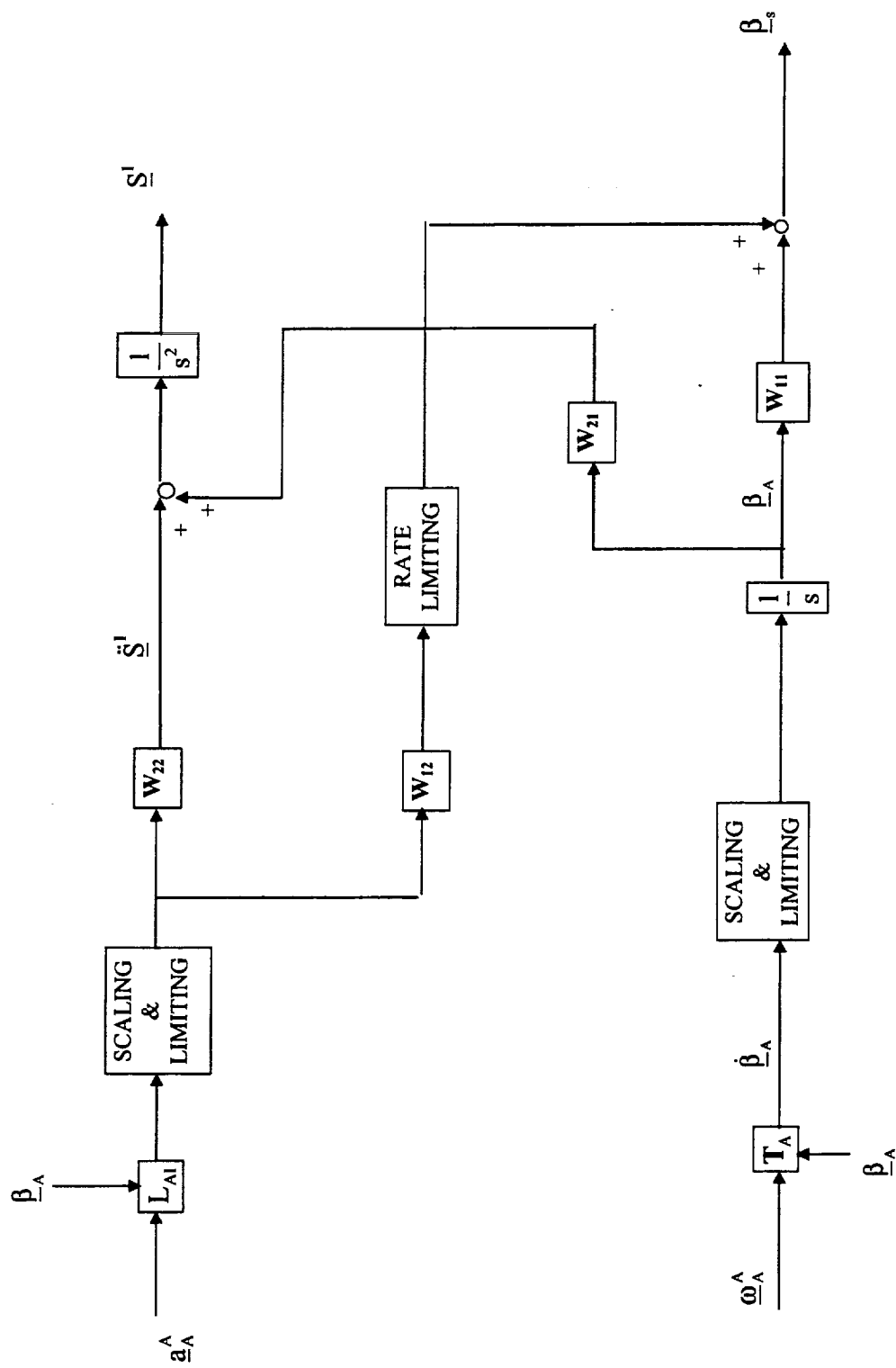


Figure 4.4.5. Optimal Washout Algorithm.

This Page Intentionally Left Blank.

5. Changes Made in Project to Original Algorithms

For evaluation purposes, changes were made to the four algorithms. The NASA adaptive algorithm and the three washout algorithms implemented by UTIAS (classical, adaptive, and optimal) are accommodated in one program as described in Section 3. Some of the subroutines in the NASA software have functions overlapping with the functions of some subroutines in the UTIAS software. These overlapping subroutines are not included in the current evaluation software.

Limits on angular tilt rates were not included in the optimal and the NASA adaptive algorithm. In the current software these limits are included. The parameters for scaling translational and rotational inputs were different for each algorithm. For this project a scale factor of 1.0 is used in all algorithms for convenience of comparison between the aircraft motion and the simulator response.

When either the UTIAS adaptive or NASA adaptive algorithm was run, convergence to steady state oscillations was observed under rotational input. The adaptive algorithms were modified in such a manner as to influence the response of the simulator corresponding to the aircraft rotational input. The modification does not affect the response of the simulator to the aircraft translational input. The difference is that the simulator will perform pure rotation to simulate an aircraft pure rotational input. The original algorithm generated some translational acceleration under aircraft pure rotational input and this translational acceleration is occasionally unstable. The rotational response of the simulator to rotational input was not affected. This topic is further discussed in Section 5.1.

When either the UTIAS adaptive or NASA adaptive algorithm was run, unwanted spikes on the specific force of the pilot's head occurred whenever the input contains some sharp change of translational acceleration. These spikes were in the opposite direction to the correct response direction. The UTIAS report [3]; [4] mentioned these unexpected spikes but did not explain the correct reason for their occurrence. No effort was made to eliminate or decrease the spikes. The cause for the spikes was identified to be angular acceleration when the simulator is tilting for compensating aircraft translational acceleration input. This angular acceleration can be so large that it can overpower all other simulator motions at some points in time and give the pilot a perception in the opposite direction to which is expected. The angular acceleration is limited in the current software. The spikes were effectively reduced and the overall response of the simulator did not change significantly. This topic is further discussed in Section 5.2.

A platform dynamics filter was added. A model of the actual platform dynamics is not currently available. A transfer function $\frac{\omega_n^2}{s^2 + 2\zeta\omega_n + \omega_n^2}$ is used to approximate the platform dynamics, where $\zeta = 0.707$, $\omega_n = 10 \pi$ rad/s. This filter represents the specifications for the real platform design. In the output plots in Appendix A, the notation 'desired simulator displacement' and 'desired simulator angular position' means the output of a washout filter. This output has not yet passed through the platform dynamics filter. The notation 'actual simulator displacement' and 'actual simulator angular position' means the output that has passed through the platform dynamics filter.

5.1. Instability in the UTIAS Adaptive Algorithm

In some rotational input cases, the UTIAS adaptive algorithm was unstable. In the UTIAS report, Volume 2, [4] the instability of the adaptive algorithm was mentioned but the reason was not discussed. The report suggested restricting the adaptive parameters to eliminate the instability. But this restriction will make the algorithm lose some of its adaptive characteristics, i.e., the adaptive algorithm becomes less 'adaptive', and this restriction may not eliminate the instability completely in some cases. It is necessary to find the cause of the instability before trying to eliminate it.

As shown in Figure 4.3, the algorithm tries to duplicate \underline{f}_A^A in the simulator frame. The simulator rotates at an angle θ_s to simulate an aircraft rotation of θ_A . It is always true that $|\theta_s| < |\theta_A|$ unless both are zero. In the pure rotational input case in which $\underline{a}_A = 0$, the aircraft and simulator angular positions are plotted in Figure 5.1 below:

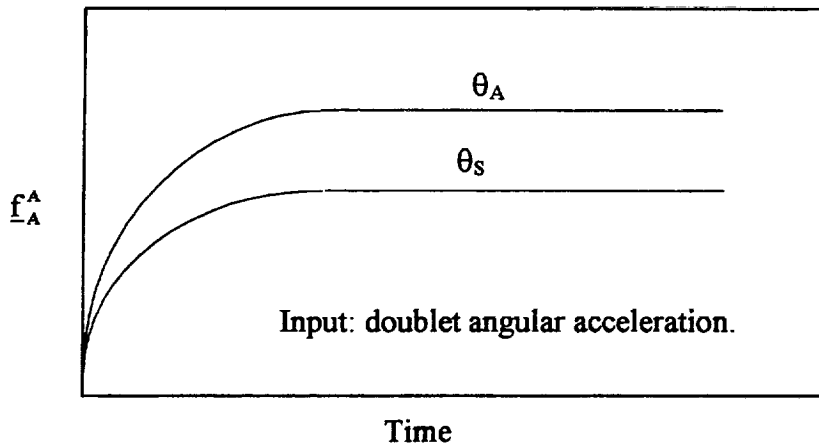


Figure 5.1. Aircraft and Simulator Angular Position under Pure Rotational Input

It is observed that with the doublet angular acceleration input, both the aircraft and the simulator angular positions reach steady state after some time. In these steady

states, \underline{g} imposes a force \underline{g}^A on the pilot's head in the aircraft and a force \underline{g}^S on the pilot's head in the simulator, where $\underline{g}^A \neq \underline{g}^S$. For example, in the pitch test case,

$$\begin{aligned}\underline{g}^A &= [-g * \sin \theta_A \quad 0 \quad g * \cos \theta_A]^T \\ \underline{g}^S &= [-g * \sin \theta_s \quad 0 \quad g * \cos \theta_s]^T\end{aligned}\tag{5-1}$$

with $\underline{f}_{2x}^I = \underline{g}^A - \underline{g}^S = [-g * (\sin \theta_s - \sin \theta_A) \quad 0 \quad g * (\cos \theta_s - \cos \theta_A)]^T$.

The UTIAS adaptive algorithm attempted to compensate for \underline{f}_{2x}^I by passing $\underline{\ddot{S}}^I$ and $\dot{\beta}_{st}$ through both the translational and crossover tilt channels. Now look at the formula for generating the simulator translational acceleration:

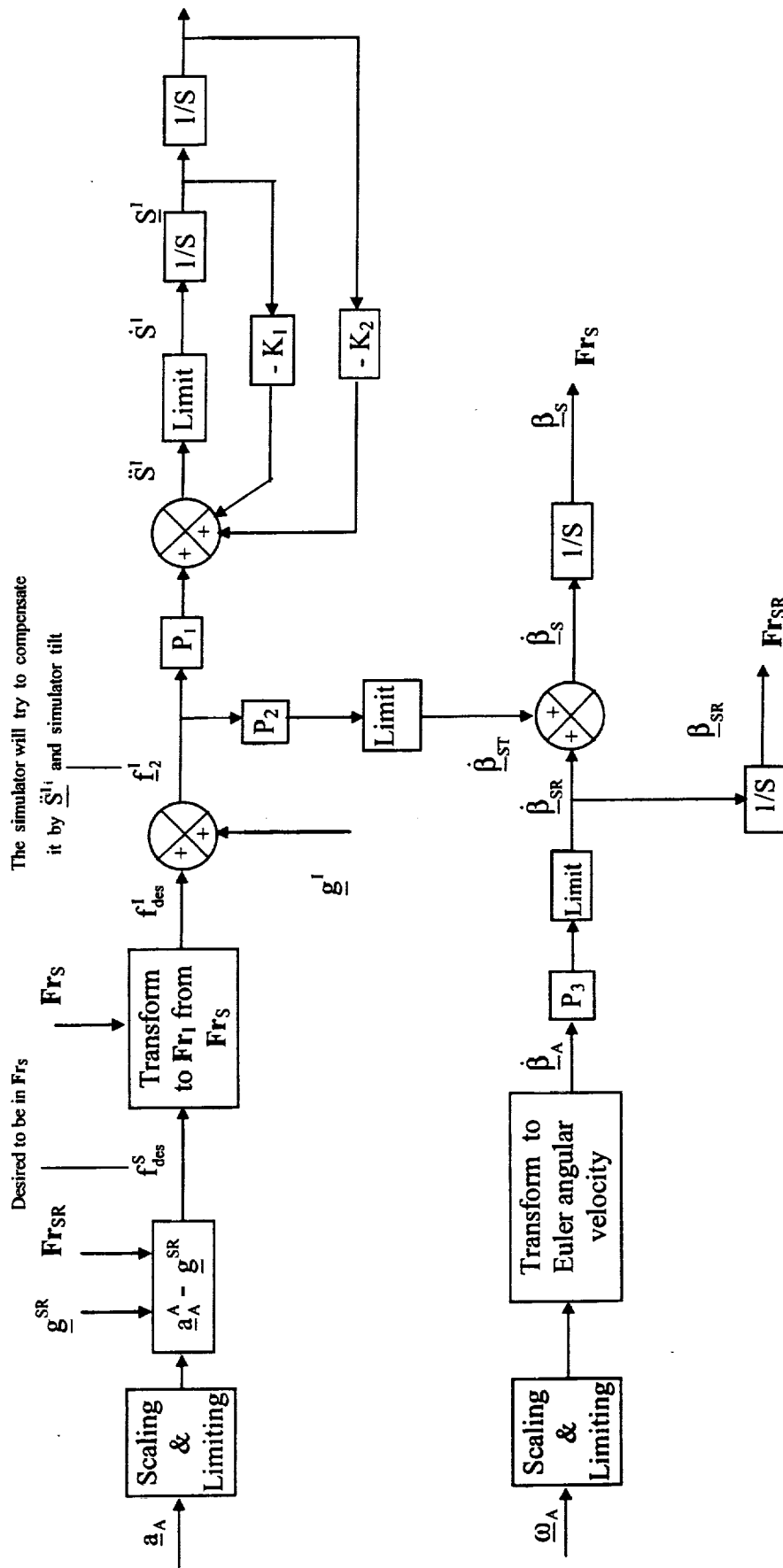
$$\underline{\ddot{S}}_x^I = p_{x1} * \underline{f}_{2x}^I - k_{x1} * \underline{S}_x^I - k_{x2} * \dot{\underline{S}}_x^I\tag{5-2}$$

where k_{x1} and k_{x2} are constant scalars and p_{x1} is an adaptive parameter. \underline{f}_{2x}^I is not zero in steady state since $|\theta_s| < |\theta_A|$. Since all washout algorithms should attempt to wash out \underline{S}^I to zero in steady states, the adaptive algorithms would attempt to bring the simulator to its neutral position in the steady state. When the simulator reaches its neutral position, i.e., $\underline{S}_x^I = 0$, it holds that $\underline{\ddot{S}}_x^I + k_{x2} * \dot{\underline{S}}_x^I = p_{x1} * \underline{f}_{2x}^I$. Since \underline{f}_{2x}^I is not equal to zero, $\underline{\ddot{S}}^I$ and $\dot{\underline{S}}^I$ cannot both be zero. Then the simulator cannot stay in the neutral position, i.e. it must continue moving. Because the adaptive algorithms are attempting to return the simulator to its neutral position in the steady state and the simulator cannot stay in its neutral position, the simulator oscillates around its neutral position. This is the cause of the instability under rotational input. This analysis is consistent with the results generated by the original adaptive algorithms before revisions were made.

The difference between \underline{g}^A and \underline{g}^S under rotational input in fact cannot be eliminated. It cannot be eliminated by simulator rotation since $|\theta_s| < |\theta_A|$. Neither can it be eliminated by $\ddot{\underline{S}}^I$ because $\ddot{\underline{S}}^I$ cannot be sustained for a long time and \underline{f}_{2x}^I is a long-term force. From another point of view, since θ_A is simulated by θ_s and \underline{g}^A and \underline{g}^S are just \underline{g}^I rotated by $\underline{\beta}_A$ and $\underline{\beta}_s$ respectively, it is reasonable to simulate \underline{g}^A by \underline{g}^S .

The block diagram for the revised algorithm is shown in Figure 5.2. \underline{g}^A is no longer followed but \underline{g}^{SR} is used as direct input. \underline{g}^{SR} is \underline{g} in an imaginary reference frame Fr_{SR} . If the simulator only responds to angular input, the simulator frame is Fr_{SR} . In the pure rotational case, $Fr_{SR} = Fr_s$, then $\underline{g}^{SR} = \underline{g}^S$.

The original algorithms had an active translational and tilt channel under pure rotational input. Neither $\dot{\underline{\beta}}_{ST}$ nor $\ddot{\underline{S}}^I$ was zero and it was often unstable under pure rotational input or a mixture of translational and rotational input. The revised algorithm generates the same results as the results generated by the original adaptive algorithms under pure translational input. The difference is that the revised algorithm has a null translational channel under pure rotational input. Both $\dot{\underline{\beta}}_{ST}$ and $\ddot{\underline{S}}^I$ are zero. The revised algorithm is stable under translational, rotational, or a mixture of translational and rotational inputs.



Note: P_1, P_2, P_3 are the adaptive parameters.

Figure 5.2. Revised UTIAS Adaptive Algorithm.

5.2. Spikes in the Outputs of the NASA and UTIAS Adaptive Algorithms

Some significant spikes occurred when either the NASA Adaptive algorithm or the UTIAS adaptive algorithm was run as shown in Figure 5.4. Whenever there was a sharp change in the translational acceleration input, a spike would occur. In the UTIAS report [1], [2], the spikes were said to be due to the attenuation of the second pulse of \underline{a}_s (simulator acceleration). But after careful examination, the real cause of the spikes is identified to be the rotational acceleration of the simulator. Since the pilot's head is in a position some distance away from the centroid of the simulator (the rotational center in the adaptive algorithms), a simulator rotational acceleration will generate some additional acceleration at the pilot's head. It is this acceleration that caused the spikes.

The specific force on the pilot's head can be expressed as:

$$\underline{f}_p = \underline{a}_{ps}^s - \underline{g}^s \quad (5-2-1)$$

where \underline{a}_{ps}^s is the acceleration at the pilot's head. \underline{a}_{ps}^s is then expressed as

$$\underline{a}_{ps}^s = \underline{a}_s^s + \ddot{\underline{R}}_s + 2 \underline{\omega}_s^s \times \dot{\underline{R}}_s + \dot{\underline{\omega}}_s^s \times \underline{R}_s + \underline{\omega}_s^s \times (\underline{\omega}_s^s \times \underline{R}_s) \quad (5-2-2)$$

where \underline{R}_s is the vector from the centroid of the simulator to the pilot's head.

Assuming $\ddot{\underline{R}}_s = \dot{\underline{R}}_s = 0$, then

$$\underline{a}_{ps}^s = \underline{a}_s^s + \dot{\underline{\omega}}_s^s \times \underline{R}_s + \underline{\omega}_s^s \times (\underline{\omega}_s^s \times \underline{R}_s) \quad (5-2-3)$$

Equation (5-2-3) indicates that both $\underline{\omega}_s^s$ and $\dot{\underline{\omega}}_s^s$ contribute to \underline{a}_{ps}^s . The limits for tilting $\underline{\omega}_s^s$ are set to 0.0524 rad/sec. Usually each component of \underline{R}_s is less than 2 m for most simulators. Therefore each component of $\underline{\omega}_s^s \times (\underline{\omega}_s^s \times \underline{R}_s)$ has a magnitude less than

0.05 m/sec². This term will not contribute much to \underline{a}_{PS}^s . Major attention will be paid to $\underline{\dot{\omega}}_s^s \times \underline{R}_s$, which is expressed as:

$$\begin{aligned}\underline{\dot{\omega}}_s^s \times \underline{R}_s &= (\dot{p} \hat{i} + \dot{q} \hat{j} + \dot{r} \hat{k}) \times (R_{sx} \hat{i} + R_{sy} \hat{j} + R_{sz} \hat{k}) \\ &= (R_{sz} \dot{q} - R_{sy} \dot{r}) \hat{i} + (R_{sx} \dot{r} - R_{sz} \dot{p}) \hat{j} + (R_{sy} \dot{p} - R_{sx} \dot{q}) \hat{k}\end{aligned}\quad (5-2-4)$$

There are two ways to prove that $\underline{\dot{\omega}}_s^s \times \underline{R}_s$ is the cause for those spikes. First, the magnitude of $\underline{\dot{\omega}}_s^s \times \underline{R}_s$ can be estimated by hand calculation. Then the value can be compared with the magnitude of the spike. The second approach is to set $\underline{R} = (0,0,0)$, when the spikes are supposed to be eliminated completely. These two tests were performed and proved that $\underline{\dot{\omega}}_s^s \times \underline{R}_s$ is the real cause of the spikes.

Since the spikes are generated by $\underline{\dot{\omega}}_s^s \times \underline{R}_s$, there are two ways to decrease the spikes. The first is to reduce $\underline{\dot{\omega}}_s^s$ and the second is to reduce \underline{R}_s . \underline{R}_s is determined by the position of the pilot's head relative to the position of the center of the simulator rotation. Only the former can be reduced. In the UTIAS report [3], [4] the center of rotation is selected to overlap with the centroid of the simulator for minimum extensions of actuator legs. This selection is quite reasonable and any change will increase the actuator extensions. Then the only approach left is to reduce $\underline{\dot{\omega}}_s^s$.

The generation of $\underline{\dot{\beta}}_A$ was investigated as shown in Figure 5.2. The Pitch/Surge filter was examined as an example. The Roll/Sway filter has similar characteristics. In the Pitch/Surge filter,

$$q = \dot{\theta}_s = px2 * \underline{f}_{2x}^I + px3 * \dot{\theta}_s \quad (5-2-5)$$

A rotational acceleration will be generated by:

$$\dot{q} = \ddot{\theta}_s = p_{x2} * (\underline{f}_{2x}^I(i+1) - \underline{f}_{2x}^I(i)) / dt, \quad (5-2-6)$$

where i indicates the current time step and $i+1$ the next time step. If there is a sudden change in \underline{f}_{2x}^I , an extremely large $\ddot{\theta}_s$ might be generated. Furthermore, the smaller the step size dt is chosen, the larger $\ddot{\theta}_s$ will be generated. For example, if the longitudinal acceleration input is a step or a pulse input, and when $t = 0$, $\dot{\theta}_s = 0$, then at the first time step $\ddot{\theta}_s = p_{x2} * \underline{f}_{2x}^I / dt$. The UTIAS adaptive algorithm set $p_{x2} = 0.12$ initially and $dt = 0.05$ sec. Then an input which has an amplitude of 1 m/sec^2 would generate $\ddot{\theta}_s = 0.12 * 1 / 0.05 = 2.4 \text{ rad/sec}^2$ at the first time step. The UTIAS simulator used $\underline{R} = -0.2 \hat{i} - 0.465 \hat{j} - 1.783 \hat{k}$. Then $\dot{q} = \ddot{\theta}_s = 2.4 \text{ rad/sec}^2$ would generate:

$$\begin{aligned} \underline{\omega}_s^s \times \underline{R}_s &= (R_{sz}\dot{q} - R_{sy}\dot{r})\hat{i} + (R_{sx}\dot{r} - R_{sz}\dot{p})\hat{j} + (R_{sy}\dot{p} - R_{sx}\dot{q})\hat{k} \\ &= R_{sz}\dot{q}\hat{i} - R_{sx}\dot{q}\hat{k} \\ &= -1.783 \cdot 2.4 \hat{i} + 0.2 \cdot 2.4 \hat{k} \\ &\approx -4.3 \hat{i} + 0.48 \hat{k} \text{ (m/sec}^2\text{)} \end{aligned}$$

$\underline{\omega}_s^s \times \underline{R}_s$ contains a component with a magnitude of about 4 m/sec^2 in the x direction at the pilot's head. This acceleration will overpower all other simulator motion effects and give the pilot a perception in the wrong direction at the beginning of the simulation. This significantly wrong perception will also happen whenever there is a sudden change in aircraft translational acceleration.

A limit on tilting angular acceleration $\underline{\omega}$ of 0.5 rad/sec^2 has been added in the current software. Spikes were attenuated significantly at the price of a slower simulator tilt response as shown in Figure 5.5. In the former example, if the limit of $\ddot{\theta}_s$ is set to be 0.2 rad/sec^2 , then the magnitude of the spikes can then be attenuated by about ten times.

At the same time, about $10 * dt$ is needed to get $\dot{\theta}_s$ to reach the expected value. Then at about 0.5 seconds, the simulator response is slower than before the new limit is added. Since it only influences about 0.5 seconds duration, this is more tolerable than the spikes, which completely reversed the pilot's perception. On the other hand, simulator translational motion performs a more important role at the beginning of simulating a translational acceleration change. This makes the slower response more tolerable.

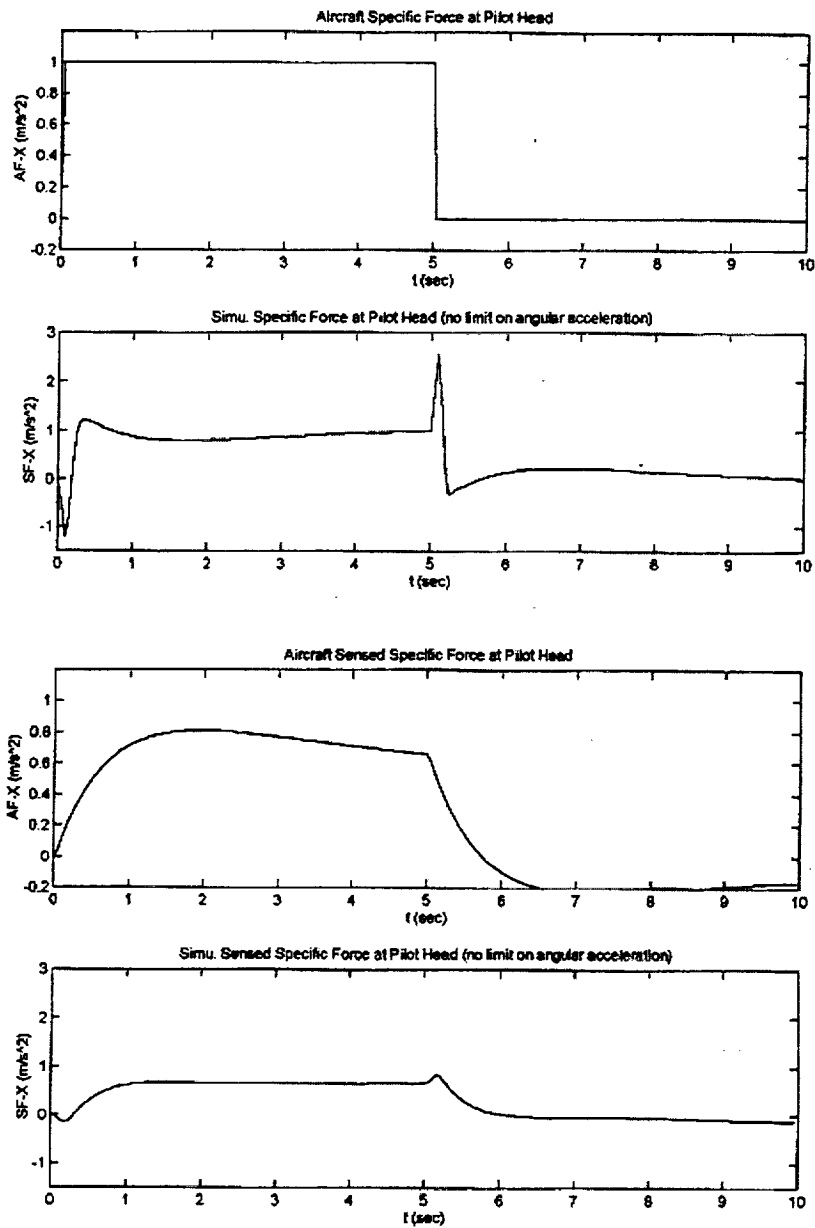


Figure 5.4. Specific Forces Generated by the NASA Adaptive Algorithm with No Limit on Angular Acceleration.

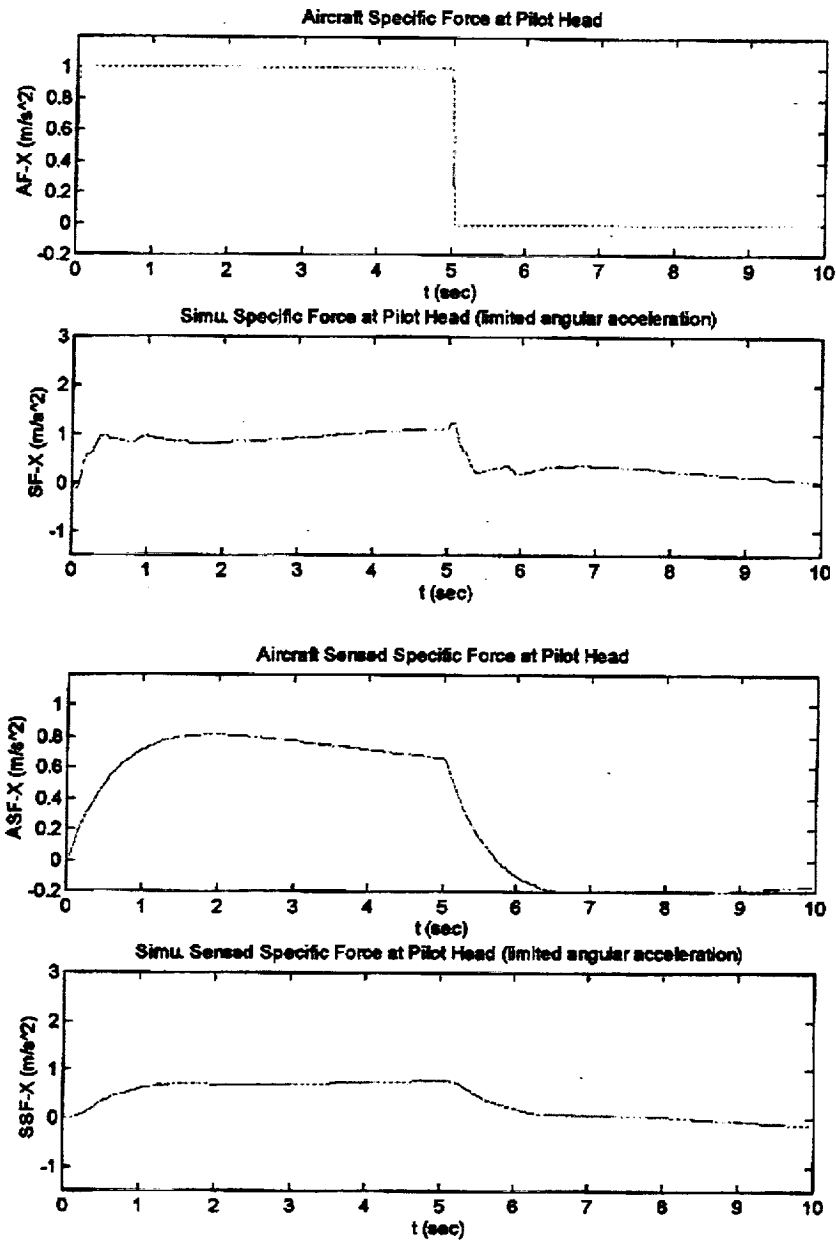


Figure 5.5. Specific Forces Generated by the NASA Adaptive Algorithm With 0.5 rad/sec/sec Limit on Angular Acceleration.

6. Phase 1 Results

6.1. Translational Input Case

Both a pulse input of 1 m/sec^2 for a 10 second duration and a sinusoidal input of 1 m/sec^2 with a frequency of 3 rad/sec were used as translational inputs. In each case the input was applied individually to each translational degree of freedom (surge, sway, and heave). The pulse contains both high frequency and low frequency components. Figure 6.1 shows the specific force at the pilot's head for the pulse input, and Figure 6.2 shows the response for the sinusoidal input.

The response onsets, i.e. the start of a response, generated by each algorithm are significantly different. During the response-sustained period the difference among those results is not significant. Major attention will be given to the response onsets. For comparison of the response onsets Figures 6.1 through 6.3 were generated with a shorter time axis (5 seconds).

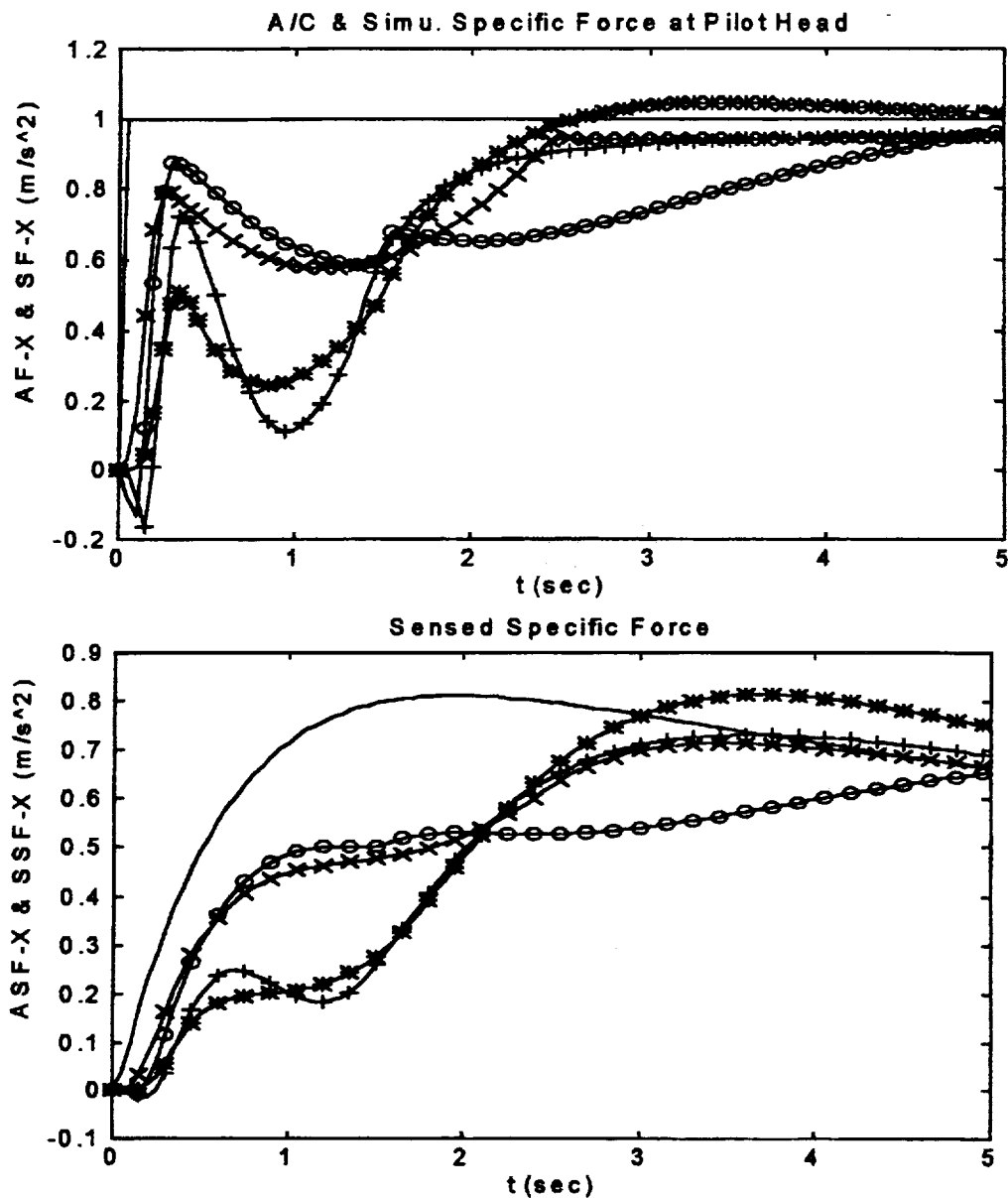
As observed in Figure 6.1 the classical algorithm and the UTIAS adaptive algorithm response onsets have smaller magnitudes and larger phase lag as compared to the NASA adaptive and optimal algorithm response onsets. If the responses within the first second are inspected, the classical algorithm generates a response with a magnitude equal to about one half the magnitude of the aircraft response. The other algorithms generate responses with higher magnitude. Note that for an easy comparison the input scaling gain has been set to one for all algorithms. The optimal algorithm had the least onset lag and the NASA adaptive algorithm had the second least onset lag, with the classical algorithm and the UTIAS adaptive algorithm having larger lags by about 0.1 to 0.2 seconds. The "sags" following the onsets generated by the classical algorithm and the

UTIAS adaptive algorithm were significantly larger than the sags generated by the optimal and the NASA adaptive algorithm. Large sags usually decrease the fidelity of motion cues.

The response offsets are similar to the response onsets. It should be noted that the UTIAS adaptive algorithm has some excessive delay. This delay is due to the tilt adaptive parameter p_2 being driven to a very small value after the input magnitude is sustained for several seconds. The NASA adaptive algorithm has a fixed tilt channel parameter k_3 so that it avoids the extra delay.

These differences on response onsets and offsets might have significant meaning in the overall fidelity of those algorithms. For further comparison, in Figure 6.2 the specific force at the simulator pilot's head corresponding to sinusoidal input was generated. It is observed that the NASA adaptive algorithm and the optimal algorithm generated responses with high magnitude. The UTIAS adaptive algorithm generated response with medium magnitude and the classical algorithm generated response with low magnitude. The optimal algorithm has the highest fidelity in shape. All other algorithms generated some extra extremum points. The NASA adaptive and UTIAS adaptive algorithms generated specific forces in a wrong direction during about the beginning 0.3 seconds. This problem on specific force in a wrong direction was discussed in Section 5.1.

On the other hand, the optimal algorithm and NASA adaptive algorithm resulted in significantly larger actuator extension costs than the classical algorithm and the UTIAS adaptive algorithm.



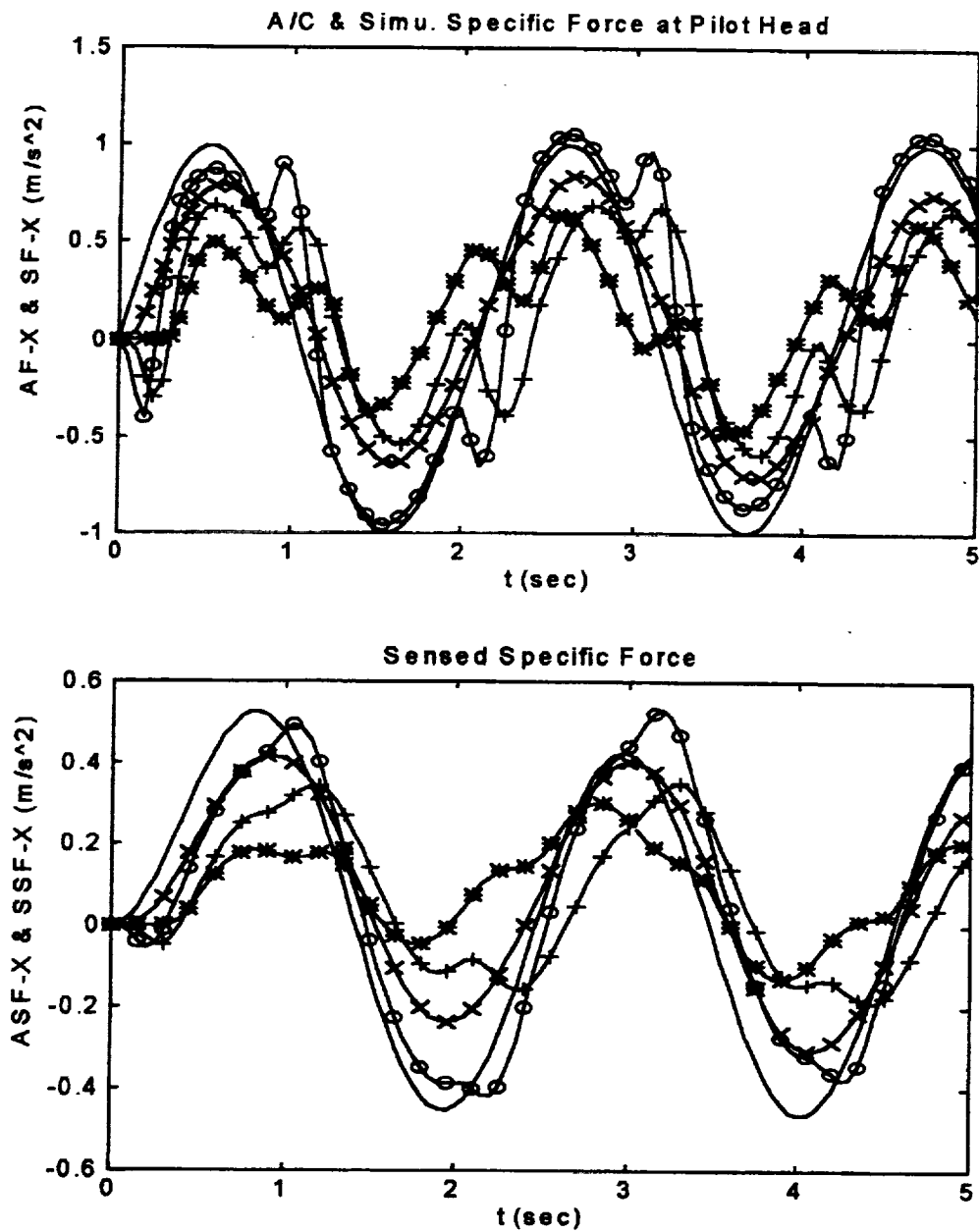
Line with no marks -- Aircraft response.

Line with marks -- Simulator response.

*-- Classical o-- NASA Adaptive +-- UTIAS Adaptive x-- Optimal

Input: pulse x-acceleration input, magnitude = $1.0 m/sec^2$, duration = 10 seconds.

Figure 6.1. Specific Force at Pilot's Head Corresponding to Pulse Input.



Line with no marks -- Aircraft response.

Line with marks -- Simulator response.

*-- Classical o-- NASA Adaptive +-- UTIAS Adaptive x-- Optimal

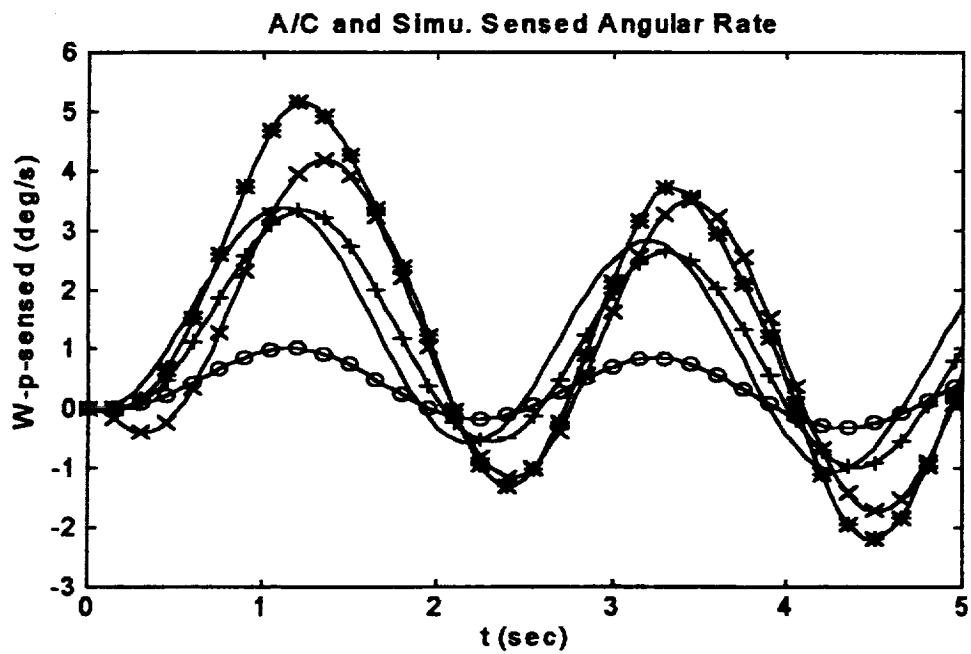
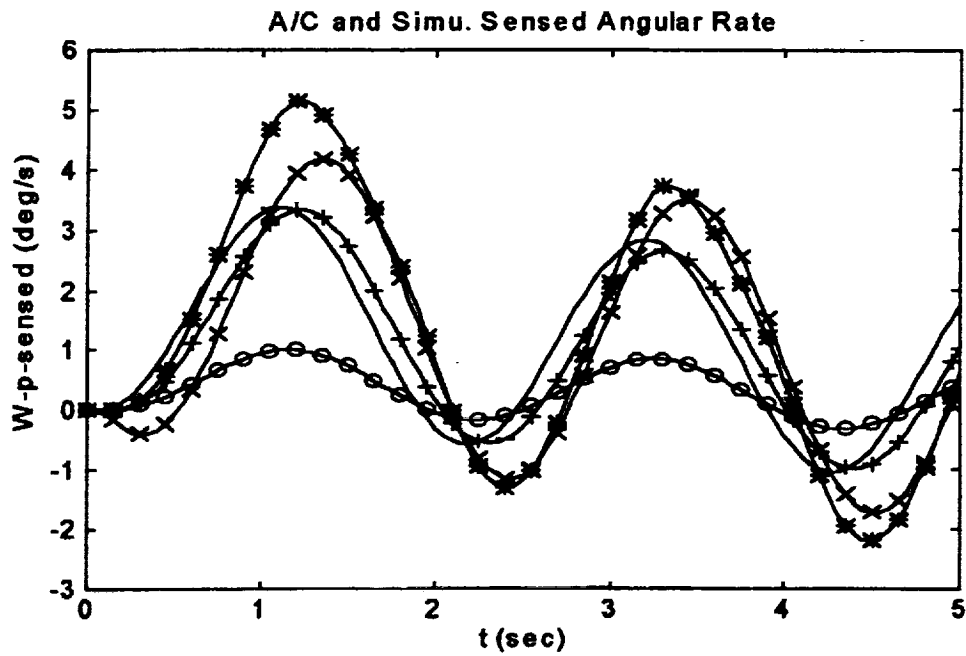
Input: sinusoidal x-acceleration input, magnitude = 1.0 m/sec^2 , $\omega = 3 \text{ rad/sec}$.

Figure 6.2. Specific Force at Pilot's Head Corresponding to Sinusoidal Input.

6.2. Rotational Input Case

A doublet pulse angular acceleration input of 0.05 rad/sec^2 for a 10 second duration was used. The classical angular rate response has some overshoot when it drops. This causes the simulator angular position response to have a slight unwanted drop (Appendix B Figure B.1). The amplitude of the angular position generated by the optimal algorithm slightly exceeds the aircraft angular position (Appendix B Figure B.2). The UTIAS adaptive algorithm and the NASA adaptive algorithm both have smooth angular position response. The magnitudes of the simulator angular position responses are significantly attenuated, especially in the NASA adaptive algorithm (Appendix B Figures B.3 and B.4). In the UTIAS Report Volume 2 [4], the Bode frequency responses of the classical algorithm and optimal algorithm were presented. These plots show that the gain of the classical and optimal response fluctuates around one and some small phase shift would happen in both the classical algorithm and optimal algorithm. These analyses are consistent with the graphs in Appendix B. For further comparison some sinusoidal input cases were studied. From Figure 6.3 it can be seen that in some input cases the gain of the classical and optimal algorithms could exceed one. The phase shifts of the classical algorithm and optimal algorithm are larger than the phase shifts of the UTIAS adaptive algorithm and NASA adaptive algorithm. The NASA adaptive algorithm has a response with much smaller magnitude.

The optimal algorithm resulted in the largest jack extension costs. The UTIAS adaptive algorithm and the NASA adaptive algorithm resulted in jack extension costs smaller than the classical algorithm and the optimal algorithm.



Line with no marks -- Aircraft response.

Line with marks -- Simulator response.

*-- Classical o-- NASA Adaptive +-- UTIAS Adaptive x-- Optimal

Input: Sinusoidal roll acceleration input, magnitude = 0.05 rad/sec^2 , $\omega = 3 \text{ rad/sec}$.

Figure 6.3. Aircraft and Simulator Angular Rate Response.

6.3. Conclusions and Recommendations

At the current stage, in translational input cases the optimal algorithm and the NASA adaptive algorithm perform better than the classical algorithm and the UTIAS adaptive algorithm. In rotational input cases, the UTIAS adaptive and the NASA adaptive algorithm perform better than the classical and the optimal algorithm.

The classical algorithm is a type of linear algorithm distinguished by the fact that the washout output is proportional to the input. This is an undesirable attribute because the duration of the onset cue is then limited by the maximum amplitude input case. Therefore in lower amplitude case the onset cue cannot be sustained any longer even though the hardware motion resource is quite available. The classical algorithm is therefore not recommended.

The NASA adaptive algorithm has been improved by eliminating unwanted spikes and excessive oscillations. The NASA adaptive algorithm currently performs well.

The UTIAS adaptive algorithm is expected to perform better than it does currently. It has also been improved by eliminating unwanted spikes and excessive oscillations. Some more improvements are needed such as increasing the slow tilt response by fixing or restricting the tilt adaptive parameters, thus changing some parameters to make the translational response be sustained for a longer time. With these further improvements implemented, the UTIAS adaptive algorithm might be attractive.

For the optimal algorithm, if selecting another washout frame can decrease the actuator extensions, this algorithm will be very attractive. Currently the optimal

algorithm performs washout in the pilot head frame. As discussed in the UTIAS report, this frame selection results in some extra jack extension costs when the simulator rotates.

A washout frame whose origin lies on the simulator centroid is desired. But the desired frame will cause some cross coupling in the pilot specific force sensation. This cross coupling might make the generation of the optimal transfer functions much more complicated or even not practically feasible. Further study on the optimal washout in the simulator centroid frame is recommended.

7. Phase 2 Optimal Algorithm Redevelopment

In Phase 2, further comparisons are made between the NASA adaptive algorithm and a redeveloped optimal algorithm. The classical algorithm will not be evaluated due to the undesirable results obtained in Phase 1. Since the adaptive algorithms are fundamentally similar with the NASA adaptive algorithm having more desirable features than the UTIAS algorithm, the UTIAS adaptive algorithm will not be evaluated in Phase 2.

The optimal algorithm will be revised with the center of rotation redefined and utilizing a new cost function with additional terms. In addition, a new design approach that integrates FORTRAN/MATLAB/SIMULINK is developed in which a set of selected parameters can be tested in 30 seconds as compared to 15 minutes with a conventional approach. The revised optimal algorithm also includes a new vestibular model.

Two general problems that were unresolved in Phase 1 are addressed in Phase 2. The first problem was that for inputs with large magnitudes all the algorithms tended to drive the simulator beyond its motion limit. A nonlinear gain algorithm was developed so that the simulator would not reach its motion limit for most input cases. The second problem was that when the simulator reached its limit an algorithm was needed to brake the simulator to a full stop and release the brake at a proper time to allow the simulator to again follow the washout algorithm output. For this problem a braking algorithm was developed in Phase 2.

7.1. New Vestibular Models

The semicircular canals have been well studied by many researchers. A model that was more consistent with the results of recent research could be found through a review of the reports written by those researchers. The construction and justification of this proposed model is discussed in detail in Appendix A. The revised model for practical usage (neglecting the short time constant) can be expressed as:

$$\frac{\hat{\underline{\omega}}}{\underline{\omega}} = \frac{\tau_p \tau_a s (1 + \tau_L s)}{(1 + \tau_a s)(1 + \tau_p s)} \quad (7-1-1)$$

where $\underline{\omega}$ is the input angular acceleration and $\hat{\underline{\omega}}$ is the sensed angular velocity. τ_a , τ_L , and τ_p are time constants. $\tau_a = 30$ sec, $\tau_L = 0.06$ sec, and $\tau_p = 5.7$ sec.

The UTLAS report [3] suggested a simplified otolith sensation model that ignored the short time constant term $(\tau_s s + 1)$ in Equation (2-1-2) for practical usage:

$$\frac{\hat{\underline{f}}}{\underline{f}} = \frac{K (\tau_a s + 1)}{(\tau_L s + 1)} \quad (7-1-2)$$

where \underline{f} is the input specific force and $\hat{\underline{f}}$ is the sensed specific force.

Human sensation models (7-1-1) and (7-1-2) are used in the revised development of the optimal algorithm.

7.2. Redefinition of Center of Rotation

In Phase 1, the optimal algorithm resulted in significantly larger actuator extensions than the other algorithms. It was found that this problem arose from the different selection of the center of rotation. If the center of rotation is selected at the centroid of the simulator, there are several cross-couplings:

pitch \rightarrow surge, pitch \rightarrow heave, roll \rightarrow sway, roll \rightarrow heave, yaw \rightarrow sway, yaw \rightarrow surge.

This can be shown in Figure 7.1:

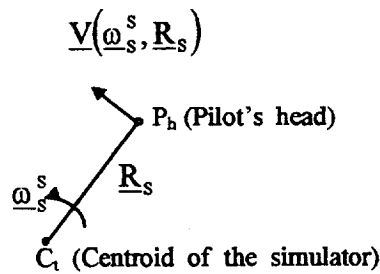


Figure 7.1. Phase 1 Optimal Algorithm Center of Rotation

An angular velocity $\underline{\omega}_s^s$ with respect to the centroid of the simulator always generates a translational motion at the pilot's head.

If the pilot's head is selected as the center of rotation and the vestibular system is considered as the only motion sensation organ of the pilot, all the cross-couplings will disappear. This is the reason for which the UTIAS report [3] chose the pilot's head as the center of simulator rotation. Unfortunately, this selection resulted in excessively large actuator extensions in some input cases. This can be shown by an example in which the simulator has both translational and rotational displacement.. For example, in the pitch/surge mode the same tilt angle θ and the same linear displacement S_x are generated in two different systems, one of which has the pilot's head and the other has the centroid of the simulator as the center of rotation. The motion of the simulator is shown in Figure

7.2. Figure 7.2 shows clearly that the different selections of center of rotation will result in different amounts of actuator extensions.

The adaptive algorithms selected the centroid of the simulator as the center of rotation. However, cross couplings were ignored in the algorithm development. Test results showed that some undesirable spikes were generated due to the cross couplings. Some effort was spent on reducing or eliminating the magnitudes of the spikes in Phase 1. The spikes were reduced but not eliminated. The spikes may generate some negative motion cue to the pilot's sensation or at least cause some delay on the pilot's sensation on the positive motion cue. In Phase 2, a new optimal algorithm that chooses the centroid of the simulator as the center of rotation was developed. Cross couplings were handled explicitly and the spikes were eliminated in the new optimal algorithm.

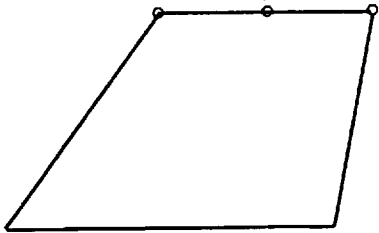
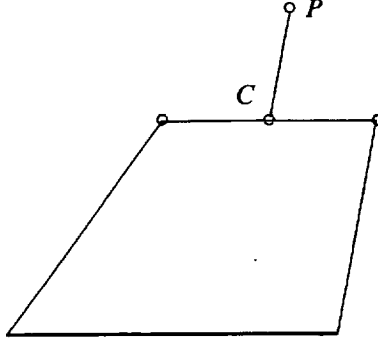
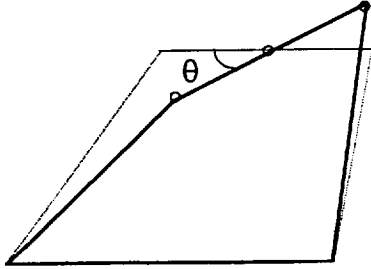
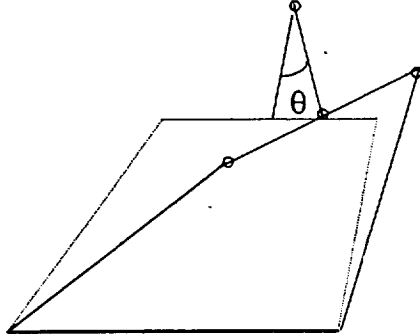
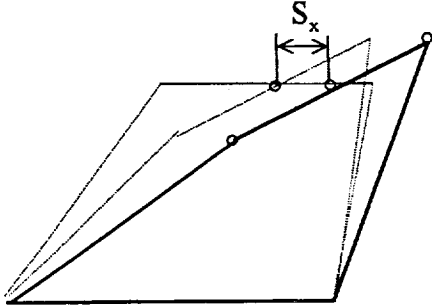
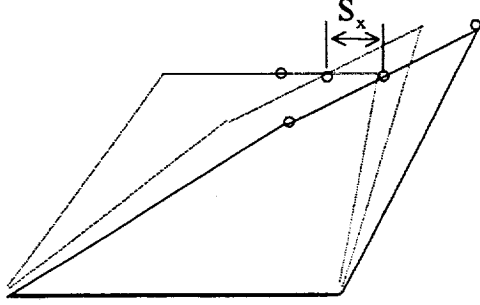
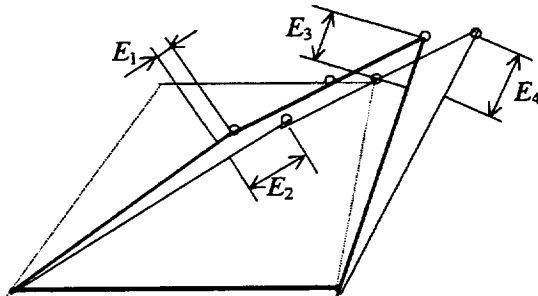
	Centroid of simulator is the center of rotation	Pilot's head is the center of rotation
Original simulator position		
With angular displacement		
With both angular and linear displacement		
Compare final actuator displacements in two different systems	 <p> E_1 : Actuator Extension relative to the original position E_1 and E_3 have the centroid of the simulator as the center of rotation E_2 and E_4 have the pilot's head as the center of rotation </p>	

Figure 7.2. Actuator Extensions for Different Centers of Rotation

7.3. Revised Development of Optimal Algorithm

7.3.1. Construction of the System Equations and Cost Function

The optimal filters for the four modes, pitch/surge, roll/sway, heave, and yaw are designed separately in the optimal algorithm development. This design technique was also employed in the development of the other three motion-base drive algorithms.

In Phase 2 only the pitch/surge mode is redefined. The roll/sway mode filter design was based on the pitch/surge redevelopment. The heave and yaw modes are unchanged from Phase 1. The revised development for the pitch/surge mode is as follows:

(a) Rotational Motion Sensation

Input:

$$\underline{u} = \begin{bmatrix} \ddot{\theta} \\ a_x \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

The semicircular canal sensation model given in Equation (7-1-1) is then used to obtain the rotational sensation \hat{q} :

$$\hat{q} = \frac{\tau_a s(1 + \tau_L s)}{(1 + \tau_a s)(1 + \tau_p s)} u_1 = \frac{T_2 s^2 + T_3 s}{s^2 + T_1 s + T_0} u_1 \quad (7-3-2)$$

where $T_0 = \frac{1}{\tau_1 \tau_a}$, $T_1 = \frac{\tau_1 + \tau_a}{\tau_1 \tau_a}$, $T_2 = \tau_L \tau_a T_0$, $T_3 = \tau_a T_0$

Now expressing the above in state space notation yields the following equations:

Define $\underline{x}_{1-2} = [x_1 \ x_2]^T$, then

$$\begin{cases} \dot{\underline{x}}_{1-2} = \underline{A}_{sc} \underline{x}_{1-2} + \underline{B}_{sc} \underline{u} \\ \hat{q} = \underline{C}_{sc} \underline{x}_{1-2} + \underline{D}_{sc} \underline{u} \end{cases} \quad (7-3-3)$$

where

$$\mathbf{A}_{ssc} = \begin{bmatrix} -T_1 & 1 \\ -T_0 & 0 \end{bmatrix}, \quad \mathbf{B}_{ssc} = \begin{bmatrix} T_3 - T_1 T_2 & 0 \\ -T_0 T_2 & 0 \end{bmatrix},$$

$$\mathbf{C}_{ssc} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \mathbf{D}_{ssc} = \begin{bmatrix} T_2 & 0 \end{bmatrix}$$

(b) Translational Motion Sensation

The otolith sensation model given in Equation (2-1-2) can be rewritten in terms of the break frequencies A_0 and B_0 :

$$\frac{\hat{\underline{f}}}{\underline{f}} = \frac{G_0(s + A_0)}{(s + B_0)} \quad (7-3-4)$$

where \underline{f} is the input specific force and $\hat{\underline{f}}$ is the sensed specific force. $G_0 = 5.86$,

$A_0 = 13.2 \text{ sec}$, $B_0 = 5.33 \text{ sec}$.

Input :

$$\underline{u} = \begin{bmatrix} \ddot{\theta} \\ a_x \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

The center of simulator rotation is set at the centroid of the simulator. Then

$$\underline{f} = a_x + g\theta - R_{sz}\ddot{\theta} \quad (7-3-5)$$

where R_{sz} is the z-component of the vector from the centroid of the simulator to the pilot's head. Equation (7-3-5) can also be expressed in the LaPlace domain:

$$\underline{f}(s) = u_2(s) + \left(g \frac{1}{s^2} - R_{sz}\right)u_1(s) \quad (7-3-6)$$

Specific force sensation:

$$\begin{aligned}
 \hat{f} &= G_0 \frac{s + A_0}{s + B_0} f \\
 &= G_0 \frac{s + A_0}{s + B_0} (u_2 + (g \frac{1}{s^2} - R_{sz}) u_1) \\
 &= G_0 \frac{s + A_0}{s + B_0} [(g \frac{1}{s^2} - R_{sz}) \quad 1] \underline{u} \\
 &= G_0 \left[\frac{-R_{sz}s^3 - R_{sz}A_0s^2 + gs + gA_0}{s^3 + B_0s^2} \quad \frac{s + A_0}{s + B_0} \right] \underline{u} \\
 &= G_0 \left[\frac{R_{sz}(B_0 - A_0)s^2 + gs + gA_0}{s^3 + B_0s^2} \quad \frac{A_0 - B_0}{s + B_0} \right] \underline{u} + G_0 [-R_{sz} \quad 1] \underline{u}
 \end{aligned} \tag{7-3-7}$$

State-space model:

Define $\underline{x}_{3-6} = [x_3 \ x_4 \ x_5 \ x_6]^T$, then

$$\begin{cases} \dot{\underline{x}}_{3-6} = \mathbf{A}_{oto} \underline{x}_{3-6} + \mathbf{B}_{oto} \underline{u} \\ \hat{\underline{f}} = \mathbf{C}_{oto} \underline{x}_{3-6} + \mathbf{D}_{oto} \underline{u} \end{cases} \tag{7-3-8}$$

where

$$\mathbf{A}_{oto} = \begin{bmatrix} -B_0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -B_0 \end{bmatrix}, \quad \mathbf{B}_{oto} = G_0 \begin{bmatrix} R_{sz}(B_0 - A_0) & 0 \\ gA_0 & 0 \\ g & 0 \\ 0 & A_0 - B_0 \end{bmatrix},$$

$$\mathbf{C}_{oto} = [1 \ 0 \ 0 \ 1], \quad \mathbf{D}_{oto} = G_0 [-R_{sz} \ 1]$$

(c) Combination of Both Rotational and Linear Motion Sensation

$$\begin{cases} \dot{\underline{x}}_{1-6} = \mathbf{A}_v \underline{x}_{1-6} + \mathbf{B}_v \underline{u} \\ \hat{\underline{y}}_1 = \mathbf{C}_v \underline{x}_{1-6} + \mathbf{D}_v \underline{u} \end{cases} \tag{7-3-9}$$

where

$$\underline{x}_{1-6} = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6]^T, \quad \hat{\underline{y}}_1 = \begin{bmatrix} \hat{q} \\ \hat{y}_1 \end{bmatrix}$$

$$\mathbf{A}_v = \begin{bmatrix} \mathbf{A}_{ssc} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{oto} \end{bmatrix}, \quad \mathbf{B}_v = \begin{bmatrix} \mathbf{B}_{ssc} \\ \mathbf{B}_{oto} \end{bmatrix}, \quad \mathbf{C}_v = \begin{bmatrix} \mathbf{C}_{ssc} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{oto} \end{bmatrix}, \quad \mathbf{D}_v = \begin{bmatrix} \mathbf{D}_{ssc} \\ \mathbf{D}_{oto} \end{bmatrix}$$

It is assumed that the same human motion sensation model can be applied to both the pilot in the aircraft and the pilot in the simulator. Define the input to the pilot in the

aircraft as $\underline{u}_a = \begin{bmatrix} \ddot{\theta}_A \\ a_{Ax} \end{bmatrix} = \begin{bmatrix} u_{A1} \\ u_{A2} \end{bmatrix}$ and the input to the pilot in the simulator as

$\underline{u}_s = \begin{bmatrix} \ddot{\theta}_s \\ a_{sx} \end{bmatrix} = \begin{bmatrix} u_{s1} \\ u_{s2} \end{bmatrix}$. To reduce the order of the system equation, define $\underline{x}_e = \underline{x}_s - \underline{x}_A$,

and then define the pilot's sensation error as $\underline{e} = \hat{\underline{y}}_{1s} - \hat{\underline{y}}_{1A}$, then

$$\begin{aligned} \dot{\underline{x}}_e &= \dot{\underline{x}}_s - \dot{\underline{x}}_A = \mathbf{A}_v \underline{x}_s + \mathbf{B}_v \underline{u}_s - (\mathbf{A}_v \underline{x}_A + \mathbf{B}_v \underline{u}_A) \\ &= \mathbf{A}_v \underline{x}_e + \mathbf{B}_v \underline{u}_s - \mathbf{B}_v \underline{u}_A \end{aligned} \quad (7-3-10)$$

$$\underline{e} = \mathbf{C}_v \underline{x}_e + \mathbf{D}_v \underline{u}_s - \mathbf{D}_v \underline{u}_A \quad (7-3-11)$$

(d) It is necessary for the control algorithm to explicitly access some motion states, such as the linear velocity and displacement of the simulator, which are desired to appear in the cost function. For this purpose some additional terms were grouped and included in the system equations.

$$\underline{x}_d = \begin{bmatrix} \iiint a_x dt^3 \\ \iint a_x dt^2 \\ \int a_x dt \\ \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \end{bmatrix} \quad (7-3-12)$$

$$\dot{\underline{x}}_d = \mathbf{A}_d \underline{x}_d + \mathbf{B}_d \underline{u}$$

where

$$\mathbf{A}_d = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B}_d = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}$$

(e) Input \underline{u} consists of filtered white noise. This can be expressed by the state-space equation

$$\begin{cases} \dot{\underline{x}}_n = \mathbf{A}_n \underline{x}_n + \mathbf{B}_n \underline{w} \\ \underline{u} = \underline{x}_n \end{cases} \quad (7-3-13)$$

where

$$\mathbf{A}_n = \begin{bmatrix} -1.0 & 0 \\ 0 & -1.0 \end{bmatrix}, \quad \mathbf{B}_n = \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}$$

(f) Assemble system equations.

The desired system equation can be obtained:

Define

$$\underline{x} = \begin{bmatrix} \underline{x}_e \\ \underline{x}_d \\ \underline{x}_n \end{bmatrix}$$

then

$$\begin{cases} \dot{\underline{x}} = \underline{A}\underline{x} + \underline{B}\underline{u}_s + \underline{H}\underline{n} \\ \underline{y} = \begin{bmatrix} \underline{e} \\ \underline{x}_d \end{bmatrix} = \underline{C}\underline{x} + \underline{D}\underline{u}_s \end{cases} \quad (7-3-14)$$

where

$$\underline{A} = \begin{bmatrix} \underline{A}_v & \underline{0} & -\underline{B}_v \\ \underline{0} & \underline{A}_d & \underline{0} \\ \underline{0} & \underline{0} & \underline{A}_n \end{bmatrix}, \quad \underline{B} = \begin{bmatrix} \underline{B}_v \\ \underline{B}_d \\ \underline{0} \end{bmatrix}, \quad \underline{H} = \begin{bmatrix} \underline{0} \\ \underline{0} \\ \underline{B}_n \end{bmatrix},$$

$$\underline{C} = \begin{bmatrix} \underline{C}_v & \underline{0} & -\underline{D}_v \\ \underline{0} & \underline{I} & \underline{0} \end{bmatrix}, \quad \underline{D} = \begin{bmatrix} \underline{D}_v \\ \underline{0} \end{bmatrix}.$$

with the cost function

$$J = E \left\{ \int_{t_0}^{t_1} (\underline{e}^T \underline{Q} \underline{e} + \underline{x}_d^T \underline{R}_d \underline{x}_d + \underline{u}_s^T \underline{R}_s \underline{u}_s) dt \right\} \quad (7-3-15)$$

The cost function implies that three variables are to be constrained: \underline{e} , \underline{x}_d , and \underline{u}_s .

\underline{e} is the sensation error. \underline{x}_d , which is a group of terms, along with \underline{u}_s define the rotational and linear motion of the simulator. The cost function constrains both the pilot sensation error and the simulator motion.

7.3.2. Transformation to Standard Form

The system equations and cost function can be transformed to the standard form by the following equations:

$$\text{System equation:} \quad \dot{\underline{x}} = \underline{A}'\underline{x} + \underline{B}\underline{u}' + \underline{H}\underline{n} \quad (7-3-16)$$

$$\text{Cost function:} \quad J = E \left\{ \int_{t_0}^{t_1} [\underline{x}^T \underline{R}_1' \underline{x} + \underline{u}'^T \underline{R}_2 \underline{u}'] dt \right\} \quad (P_1 \equiv 0) \quad (7-3-17)$$

where

$$\mathbf{G} = \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_d \end{bmatrix}, \quad \mathbf{R}_1 = \mathbf{C}^T \mathbf{G} \mathbf{C}, \quad \mathbf{R}_{12} = \mathbf{C}^T \mathbf{G} \mathbf{D}, \quad \mathbf{R}_2 = \mathbf{R} + \mathbf{D}^T \mathbf{G} \mathbf{D},$$

$$\mathbf{A}' = \mathbf{A} - \mathbf{B} \mathbf{R}_2^{-1} \mathbf{R}_{12}^T, \quad \underline{\mathbf{u}}' = \underline{\mathbf{u}}_s + \mathbf{R}_2^{-1} \mathbf{R}_{12}^T \underline{\mathbf{x}}, \quad \mathbf{R}_1' = \mathbf{R}_1 - \mathbf{R}_{12} \mathbf{R}_2^{-1} \mathbf{R}_{12}^T,$$

7.3.3. Solution of the Optimal Control Problem

The standard form solution to the optimal control problems can be applied since both the system equations and the cost function are in standard form. Therefore,

$$\underline{\mathbf{u}}' = -\mathbf{R}_2^{-1} \mathbf{B}^T \mathbf{P}(t) \underline{\mathbf{x}} \quad (7-3-18)$$

where \mathbf{P} is the solution of the algebraic Riccati equation:

$$\mathbf{0} = \mathbf{R}_1' - \mathbf{P} \mathbf{B} \mathbf{R}_2^{-1} \mathbf{B}^T \mathbf{P} + \mathbf{A}'^T \mathbf{P} + \mathbf{P} \mathbf{A}' \quad (7-3-19)$$

From (7-3-17) and the definition of $\underline{\mathbf{u}}'$, it is obvious that

$$\underline{\mathbf{u}}_s = -[\mathbf{R}_2^{-1} (\mathbf{B}^T \mathbf{P} + \mathbf{R}_{12}^T)] \underline{\mathbf{x}} \quad (7-3-20)$$

Define

$$\mathbf{F} = \mathbf{R}_2^{-1} (\mathbf{B}^T \mathbf{P} + \mathbf{R}_{12}^T) \quad (7-3-21)$$

Then

$$\underline{\mathbf{u}}_s = -\mathbf{F} \underline{\mathbf{x}} \quad (7-3-22)$$

Partition \mathbf{F} corresponding to the partition of $\underline{\mathbf{x}} = \begin{bmatrix} \underline{\mathbf{x}}_e \\ \underline{\mathbf{x}}_d \\ \underline{\mathbf{x}}_n \end{bmatrix}$:

$$\underline{\mathbf{u}}_s = -\begin{bmatrix} \mathbf{F}_1 & \mathbf{F}_2 & \mathbf{F}_3 \end{bmatrix} \begin{bmatrix} \underline{\mathbf{x}}_e \\ \underline{\mathbf{x}}_d \\ \underline{\mathbf{x}}_n \end{bmatrix} = -\begin{bmatrix} \mathbf{F}_1 & \mathbf{F}_2 \end{bmatrix} \begin{bmatrix} \underline{\mathbf{x}}_e \\ \underline{\mathbf{x}}_d \end{bmatrix} - \mathbf{F}_3 \underline{\mathbf{u}}_n \quad (7-3-23)$$

Eliminate $\dot{\underline{x}}_a$ from the system equations:

$$\begin{bmatrix} \dot{\underline{x}}_e \\ \dot{\underline{x}}_d \end{bmatrix} = \begin{bmatrix} \mathbf{A}_v & \mathbf{0} & -\mathbf{B}_v \\ \mathbf{0} & \mathbf{A}_d & \mathbf{0} \end{bmatrix} \begin{bmatrix} \underline{x}_e \\ \underline{x}_d \\ \underline{x}_n \end{bmatrix} + \begin{bmatrix} \mathbf{B}_v \\ \mathbf{B}_d \end{bmatrix} \underline{u}_s \quad (7-3-24)$$

After taking the Laplace transform of (7-3-23) and (7-3-24) and making some substitutions, the following equations are obtained:

$$\underline{u}_s(s) = \tilde{\mathbf{W}}(s) \cdot \underline{u}_a(s) \quad (7-3-25)$$

where

$$\tilde{\mathbf{W}}(s) = \begin{bmatrix} \mathbf{F}_1 & \mathbf{F}_2 \end{bmatrix} \begin{bmatrix} s\mathbf{I} - \mathbf{A}_v + \mathbf{B}_v\mathbf{F}_1 & \mathbf{B}_v\mathbf{F}_2 \\ \mathbf{B}_d\mathbf{F}_1 & s\mathbf{I} - \mathbf{A}_d + \mathbf{B}_d\mathbf{F}_2 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{B}_v(\mathbf{I} + \mathbf{F}_3) \\ \mathbf{B}_d\mathbf{F}_3 \end{bmatrix} - \mathbf{F}_3$$

Note that the filter matrix $\tilde{\mathbf{W}}(s)$ is not identical to the filter matrix $\mathbf{W}(s)$ described in Section 4 and implemented in the UTIAS report. The revised design of $\tilde{\mathbf{W}}(s)$ was based on control inputs of angular acceleration and linear acceleration, while the design of $\mathbf{W}(s)$ was based on control inputs of angular displacement and linear acceleration. A revised on-line Fortran development was not implemented; the pitch/surge and roll/sway filters $\mathbf{W}(s)$ developed in Section 4 were simply replaced with the revised filters $\tilde{\mathbf{W}}(s)$ in the Phase 1 on-line development.

7.3.4. Solution of the Washout Filter Coefficient

$\tilde{\mathbf{W}}(s)$ is solved by Matlab functions and Fortran programs. All the system matrices were constructed in Matlab functions. First, the weighing matrices \mathbf{Q} , \mathbf{R} , and \mathbf{R}_d in the cost function were selected. These matrices were adjusted several times in a trial and error procedure. Then the system matrices \mathbf{A}_v , \mathbf{B}_v , \mathbf{C}_v , \mathbf{D}_v , \mathbf{R}_1 , \mathbf{R}_2 , \mathbf{R}_{12} , \mathbf{A}' , and \mathbf{R}' were constructed in a proper order. The algebraic Riccati equation was defined in the

Matlab function. The definition was transported to a Fortran environment in which the algebraic Riccati equation would be solved by Fortran programs. The solution of the algebraic Riccati equation was transported back to the Matlab environment and F_1 , F_2 , and F_3 could then be defined.

After all matrices in (7-3-25) were defined, the solution of $\tilde{W}(s)$ still required symbolic calculation since there was the Laplace variable s . A series of fairly complicated Fortran programs were presented in the UTIAS report to solve $\tilde{W}(s)$. The solution required manual cancellation of common poles and zeros. It was found that the problem could be solved more efficiently by Matlab after a careful observation of Equation (7-3-25). Recall that a linear control system can be expressed in state-space form:

$$\begin{cases} \dot{\underline{x}} = \tilde{\mathbf{A}}\underline{x} + \tilde{\mathbf{B}}\underline{u} \\ \underline{y} = \tilde{\mathbf{C}}\underline{x} + \tilde{\mathbf{D}}\underline{u} \end{cases} \quad (7-3-27)$$

where $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, $\tilde{\mathbf{C}}$, $\tilde{\mathbf{D}}$ are system matrices.

Equation (7-3-27) determines a transfer function between \underline{y} and \underline{u} :

$$\underline{y} = [\tilde{\mathbf{C}}(s\mathbf{I} - \tilde{\mathbf{A}})^{-1}\tilde{\mathbf{B}} + \tilde{\mathbf{D}}] \underline{u} \quad (7-3-28)$$

By an observation of the similarity between (7-3-26) and (7-3-28), it was found that the state-space model that had \underline{u}_A as the input and \underline{u}_S as the output could be obtained directly:

$$\begin{cases} \dot{\underline{x}} = \mathbf{A}\underline{x} + \mathbf{B}\underline{u}_A \\ \underline{u}_S = \mathbf{C}\underline{x} + \mathbf{D}\underline{u}_A \end{cases} \quad (7-3-29)$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_v - \mathbf{B}_v \mathbf{F} & -\mathbf{B}_v \mathbf{F} \\ -\mathbf{B}_d \mathbf{F} & \mathbf{A}_d - \mathbf{B}_d \mathbf{F} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}_v (\mathbf{I} + \mathbf{F}_3) \\ \mathbf{B}_d \mathbf{F}_3 \end{bmatrix}, \quad \mathbf{C} = [\mathbf{F}_1 \quad \mathbf{F}_2], \quad \mathbf{D} = -\mathbf{F}_3$$

This state-space model could be converted to the transfer function model by the Matlab function `ss2tf` easily. A Matlab function was also written to cancel common poles and zeros and yield some transfer functions in reduced order. A simulation setup that reads in the transfer functions automatically was constructed in Simulink. The effects of $\tilde{\mathbf{W}}(s)$ are visualized with the simulation. If $\tilde{\mathbf{W}}(s)$ is not satisfactory, the procedure of the design of $\tilde{\mathbf{W}}(s)$ is repeated by selecting some new cost function matrices \mathbf{Q} , \mathbf{R} , and \mathbf{R}_d , until a satisfactory result can be finally approached. The procedure for the solution of $\tilde{\mathbf{W}}(s)$ is illustrated in Figure 7.3.

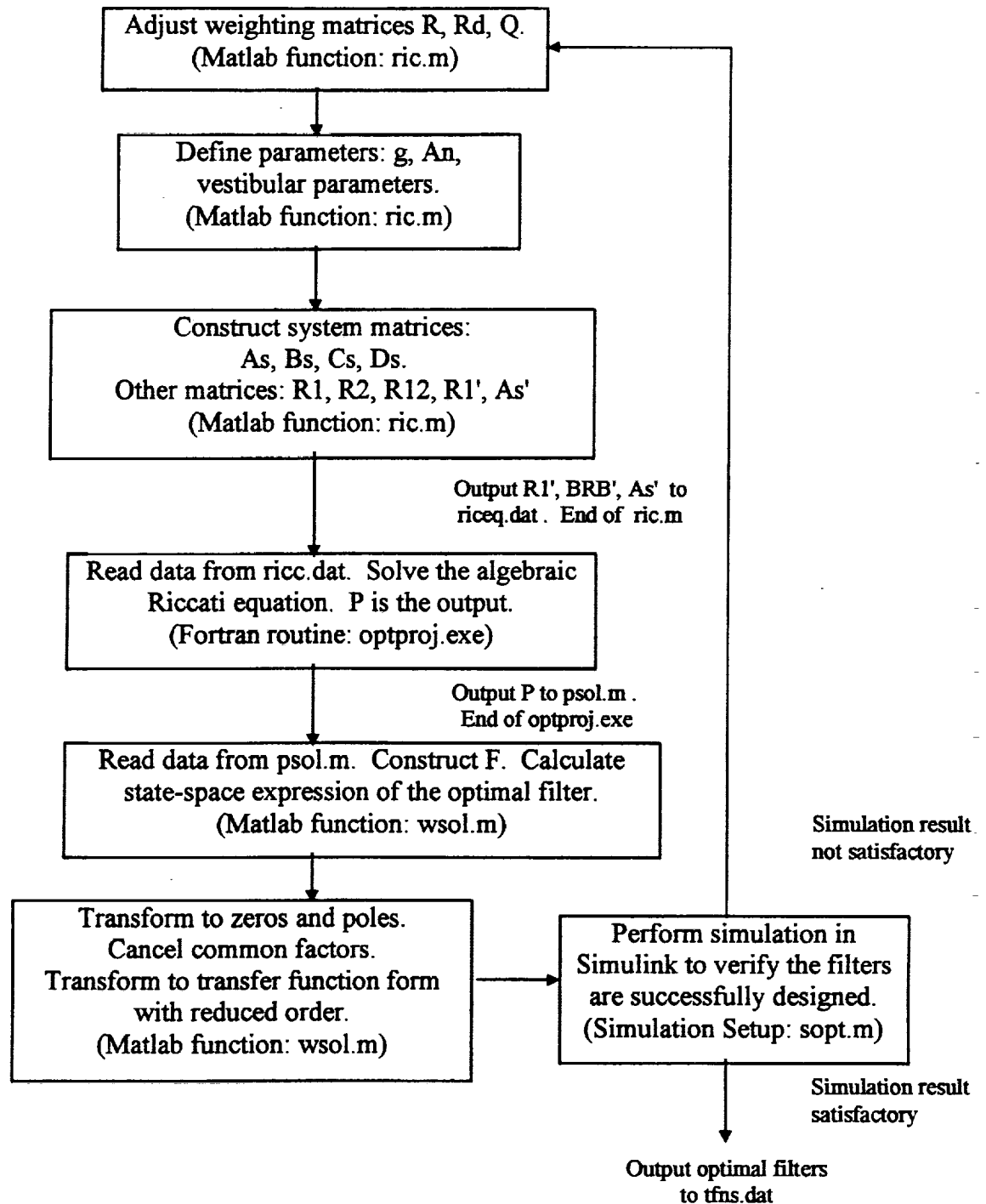


Figure 7.3. $\tilde{W}(s)$ Solution Procedure.

7.4. Nonlinear Gain Algorithm

It is desirable to maximize the magnitude of the motion cue while remaining within the operational limits of the motion system. Although it is very difficult to restrain the response within hardware limits all the time, it was found feasible to restrain the response successfully during most of the simulation time by implementing a nonlinear gain in the simulator motion-base driving software.

When the magnitude of input to the simulator motion system is small, the gain is desired to be relatively high, otherwise the output may be below the pilot's perception threshold when it should be perceptible. When the magnitude of input is high, the gain is desired to be relatively low otherwise the simulator may go beyond the hardware limits. Define the input as x and the output as y . Define x_{\max} as the expected maximum input and y_{\max} as the maximum output, and S_0 and S_1 as the slopes at $x = 0$ and $x = x_{\max}$ respectively. Four desired characteristics for the nonlinear gain can be expressed as:

- (1) $x = 0 \Rightarrow y = 0$;
- (2) $x = x_{\max} \Rightarrow y = y_{\max}$;
- (3) $y'|_{x=0} = S_0$;
- (4) $y'|_{x=x_{\max}} = S_1$;

A third order polynomial can be employed to provide functions with all the desired characteristics. This polynomial will be of the form:

$$y = c_3x^3 + c_2x^2 + c_1x + c_0$$

where

$$c_0 = 0$$

$$c_1 = s_0$$

$$c_2 = x_{\max}^{-2} \cdot (3y_{\max} - 2S_0x_{\max} - S_1x_{\max})$$

$$c_3 = x_{\max}^{-3} \cdot (S_0x_{\max} - 2y_{\max} + S_1x_{\max})$$

One example of this nonlinear gain is shown in Figure 7.4, with parameters set as $x_{\max} = 10$, $y_{\max} = 6$, $s_0 = 1.0$, $s_1 = 0.1$.

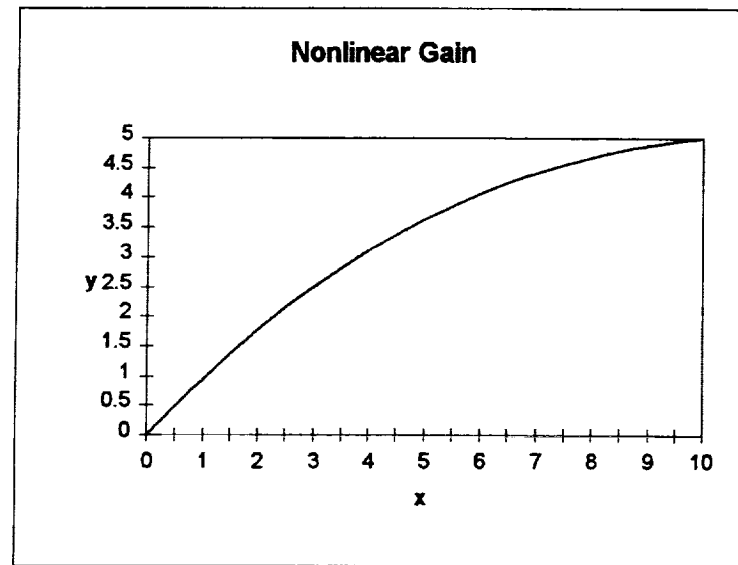


Figure 7.4. Nonlinear Gain.

The nonlinear gain developed by NASA for its motion-base drive software results in a high probability of reaching system limits for accelerations higher than 3 m/sec^2 , which is common in aircraft simulation. When the new nonlinear gain as shown in Figure 7.5 was implemented in the software, in most input cases the simulator motion was within the hardware limit.

7.5. Actuator Extension Limiting

There is a possibility that the output of the washout algorithm will drive the simulator beyond its hardware limit. It is obvious that any hardware shut down due to the simulator's excessive excursion should be avoided. This raises the requirement that the simulation software should be able to handle the situation that the simulator may need to be arrested before encountering the system limit. This necessary software subroutine is called the braking algorithm. When necessary, the braking algorithm takes over the control of the simulator from the washout algorithm. It is also desirable that when the washout algorithm would begin to drive the simulator toward smaller excursions, the braking algorithm returns control to the washout algorithm. In other words, the braking algorithm should release the system to resume regular simulation at the proper time. The logic of the braking algorithm is shown in Figure 7.6.

The algorithm makes a series of decisions to determine when to brake the simulator and while braked to determine when to release the brake. The first decision is based on an evaluation of the expression $(2 \cdot c_0 \cdot a_b \cdot s - v^2)$ for each actuator at each simulation cycle, where v is the velocity of the actuator, s is the available actuator stroke, a_b is the acceleration/deceleration by which the braking algorithm will stop the actuator, and c_0 is a coefficient less than or equal to 1 that is described below.

a_b should not be larger than the maximum actuator acceleration which is a hardware parameter. The simulation software sets a software limit position that is before the simulator hardware limit position. If the simulation cycling rate is infinitely high, when $c_0 = 1$, the actuator will be stopped exactly at the software limit position; when

$c_0 < 1$, the actuator will be stopped before the software limit position. Since the real system always has a finite cycling rate, it is possible that the actuator may go a little beyond the software limit position even though a braking algorithm is implemented. For this reason, the software limit position should be set slightly before the hardware limit position. When one actuator reaches the braking region defined by $(2 \cdot c_0 \cdot a_b \cdot s - v^2 \leq 0)$ at time t_0 , it will be decelerated by a_b . At the same time, all other actuators will be decelerated proportionally to their respective velocities at time t_0 .

The second decision is based on a comparison between the actual simulator states and the washout algorithm output. The simulator states refer to the simulator's linear and angular positions. When the simulator has been completely stopped by the braking algorithm, the comparison begins. When the washout algorithm output states are smaller than the corresponding actual simulator states, the simulator will begin to follow the washout algorithm output again.

When the braking algorithm releases the brake, the washout algorithm output may have large velocities while the simulator has small velocities at that instance. To avoid the regular simulation being resumed with any excessively large simulator acceleration, an algorithm which allows the simulator to follow the washout algorithm output gradually was designed. The algorithm expresses the linear and angular position command to the simulator as:

$$\underline{X}_{Com} = a(t) \cdot \underline{X}_{Act} + (1 - f(t)) \cdot (\underline{X}_{Wash} - \underline{X}_{Act}) \quad (7-5-1)$$

where

\underline{X}_{Com} is the commanded position which drives the simulator;

\underline{X}_{Act} is the current actual position of the simulator;

\underline{X}_{Wash} is the current output of the washout algorithm.

$a(t)$ is a function of time. It increases from 0 to $\pi / 2$ gradually.

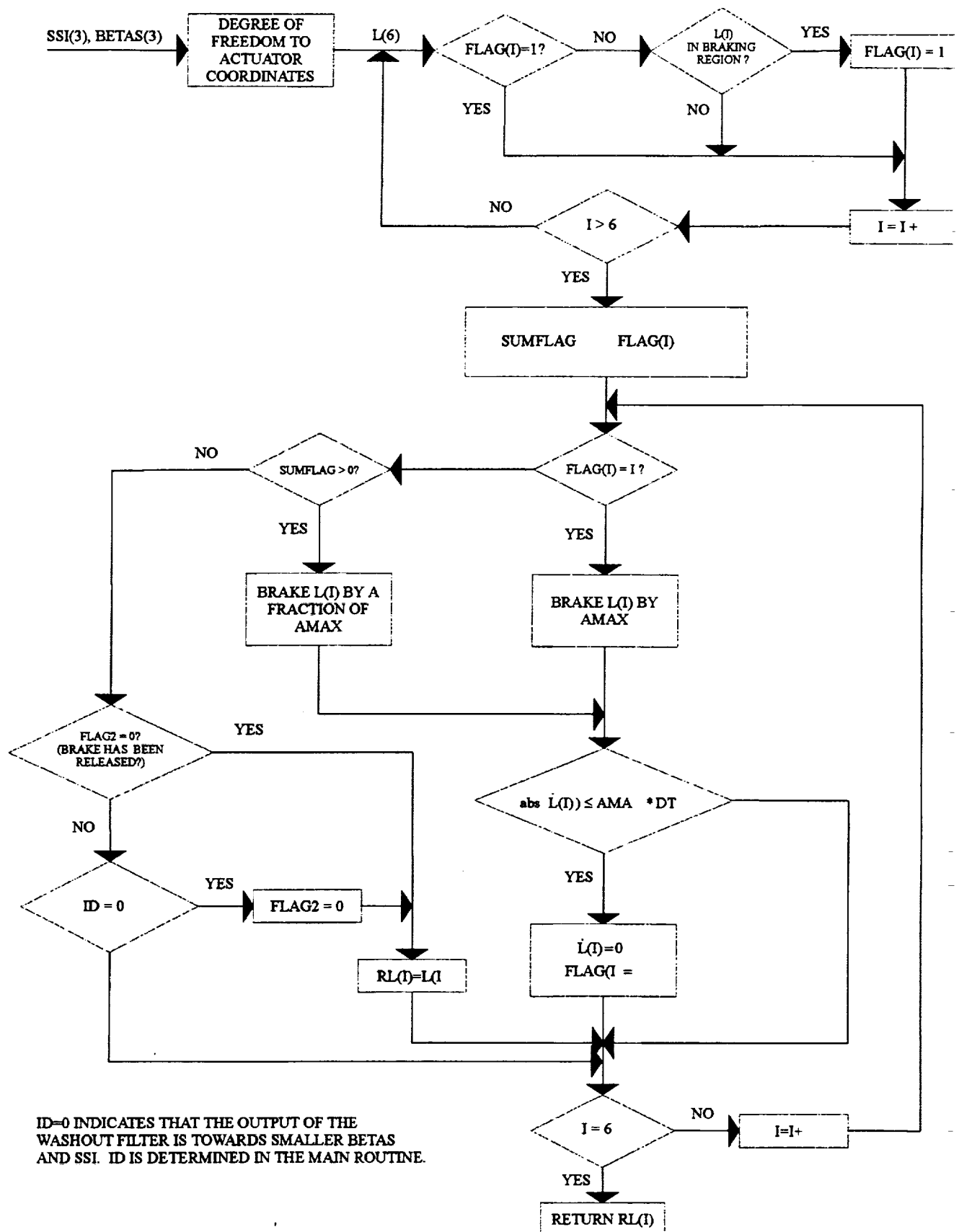


Figure 7.6. Braking Algorithm in the Actuator Driving Subroutine.

8. Phase 2 Results

8.1. Pitch/Surge Mode and Roll/Sway Mode

The pitch/surge mode and the roll/sway mode have similar characteristics with both employing the simulator tilt to provide sustained specific force cue, and both having the cross-coupling from the simulator tilt to the pilot's head translational motion. Both half cycle sinusoidal inputs and ramp to step inputs were employed to test the pitch/surge and roll/sway channel of the NASA adaptive washout algorithm and the optimal washout algorithm.

When a 0.1 Hz half cycle sinusoidal translational acceleration was employed as the input, the output of the NASA adaptive algorithm and the output of the optimal algorithm were similar. The specific forces at the simulator pilot's head generated by the two algorithms had similar magnitudes and shapes. The actuator extensions resulted from the two algorithms were also similar. One difference with the NASA adaptive algorithm was that the specific force generated went to a wrong direction at the beginning of the input and had an extra hump besides the sinusoidal hump corresponding to the half cycle sinusoidal hump of the input. The specific force output of the optimal algorithm did not have these two deviations from the shape of the input. The outputs are shown in Appendix C, Figures C.1, C.2, and C.3.

When a 0.5 Hz half cycle sinusoidal translational acceleration was used as input, the shape of the specific force at the simulator pilot's head generated by the NASA adaptive algorithm had some significant distortion. The specific force in the wrong direction at the beginning of the input had a magnitude which was about half of the magnitude of the onset of the specific force cue and a duration which was about one half

of the time by which the input reached its maximum. The specific force onset had some more wiggles than the input and the washout of the specific force had a large overshoot. Stimulated by the same input, the optimal algorithm generated a specific force at the simulator pilot's head with significantly smaller shape distortion. There was no negative cue at the beginning of the input. The overall shape of the output was similar to the input, with a small washout overshoot. The outputs are shown in Appendix C, Figures C.10, C.11, and C.12.

When a ramp to step acceleration with a ramp slope of 3 m/sec^2 was employed as the input, the specific forces at the simulator pilot's head generated by the two algorithms had similar magnitudes and shapes. The actuator extensions resulting from the two algorithms were also similar. The specific force generated by the NASA adaptive algorithm went in the wrong direction at the beginning of the input. The specific force generated by the optimal algorithm reached its maximum earlier as compared to the adaptive algorithm and then slowly decreased. The outputs are shown in Appendix C, Figures C.4, C.5, and C.6.

When a ramp to step acceleration with a ramp slope of 5 m/sec^2 was employed as the input, the shape of the specific force at the pilot's head generated by the NASA adaptive algorithm had some significant distortion. The specific force in the wrong direction at the beginning of the input had a magnitude about one half of the magnitude of the onset of the specific force cue with its duration almost equal to the input ramp duration. The onset of the specific force was followed by a sag. Stimulated by the same input, the optimal algorithm generated specific force at the pilot's head with significantly smaller shape distortion. There was no negative cue at the beginning of the input. The

overall shape of the output was similar to the input. The outputs are shown in Appendix C, Figures C.7, C.8, and C.9.

8.2. Heave Mode

A pulse acceleration with a magnitude of 3 m/sec^2 and duration of 5 seconds was employed as the input. The onset of the specific force at the simulator pilot's head generated by the NASA adaptive algorithm was similar to the one generated by the optimal algorithm. When the specific force at the aircraft pilot's head decreased, the optimal algorithm generated a corresponding drop of the specific force at the simulator pilot's head, while the NASA adaptive algorithm did not generate an obvious drop of the corresponding output. The drop of the specific force at the simulator pilot's head generated by the optimal algorithm was followed by a washout with a large overshoot. The outputs are shown in Appendix C, Figures C.13, C.14, and C.15.

8.3. Pitch Mode and Roll Mode

A doublet angular acceleration with a magnitude of 0.1 rad/sec^2 and a duration of 5 seconds was employed as the input to the pitch channel and roll channel. For the pitch test runs the shape of the specific force generated by the NASA adaptive algorithm was nearly the same as the shape of the input, while the specific force generated by the optimal algorithm slowly decreased. For the roll test run the specific force generated by the NASA adaptive algorithm was nearly the same as the shape of the input, while the specific force generated by the optimal algorithm required more time to settle during washout. Some distortion with the simulator angular velocity relative to the input was also observed with the optimal algorithm. The outputs are shown in Appendix C, Figures C.16, C.17, C.18, C.19, C.20, and C.21.

8.4. Yaw Mode

A pulse angular acceleration with a magnitude of 0.05 rad/sec^2 and duration of 5 seconds was employed as the input. Both the outputs generated by the NASA adaptive algorithm and the optimal algorithm have the characteristic that the simulator yaw rate and yaw angle were washed out toward zero when the aircraft yaw rate was rising up to a constant value and the aircraft yaw angle was always increasing. The simulator yaw rate generated by the NASA adaptive algorithm dropped much earlier as compared to the aircraft yaw rate. The optimal algorithm generated a yaw rate following the aircraft yaw rate for a longer time but also resulted in larger actuator extensions as compared to the NASA adaptive algorithm. The washout of the simulator yaw rate generated by the optimal algorithm had a larger overshoot than the one generated by the NASA adaptive algorithm. The NASA adaptive algorithm produced a distorted specific force curve with an onset greater than the aircraft which decreased too rapidly, while the optimal algorithm was closer in shape to the aircraft but with a large amount of overshoot in washout. The outputs are shown in Appendix C, Figures C.22, C.23, and C.24.

8.5. Braking Algorithm

A y-acceleration pulse with a magnitude of 10 m/sec^2 and duration of 10 seconds was employed as the input. The optimal algorithm was employed as the washout algorithm. The simulator reached its motion limit once and was successfully braked. Some large specific force spikes were generated when the simulator was being braked. The brake was then released when the washout algorithm output was driving the simulator toward smaller excursions. The outputs are shown in Figure C.25.

9. Conclusions

When the pitch/surge or roll/sway channel was tested, the specific force output of the NASA adaptive algorithm often went in the wrong direction at the beginning of a translational acceleration input. The specific force in the wrong direction resulting from the cross-coupling from the simulator tilt to the simulator pilot's head translational motion was reduced but not completely eliminated in the first phase of this project. It was found that both the magnitude and duration of the specific force in the wrong direction were still too large to be ignored when the input contained some frequency components near or higher than 0.5 Hz. As discussed in Section 1, negative motion cues should always be avoided if possible and when the motion cue onset has high importance. The specific force in the wrong direction may generate a bad motion sensation which happens at the motion onset, therefore it was desired to eliminate it completely. It was found that the optimal algorithm successfully handled this problem. The cross-coupling was explicitly expressed in the construction of the system equations when the new optimal algorithm was developed. Results showed that the specific force in the wrong direction was completely eliminated. The optimal algorithm also generated specific force outputs with significantly smaller shape distortions than the NASA adaptive algorithm did in the pitch/surge and roll/sway test cases.

In the heave test case, the NASA adaptive algorithm did not generate an obvious specific force drop when there was a large drop of the input while the optimal algorithm generated one, which was obviously desired.

When the pitch channel and the roll channel were tested, the NASA adaptive algorithm generated simulator angular velocity outputs with very nice shapes while the outputs of the optimal algorithm had some visible distortion. The optimal filters for the pitch channel were designed in the pitch/surge channel design. The filters were tuned mainly according to test runs with surge inputs. If the filters for the pitch channel are not satisfactory, it may be necessary to re-tune those pitch filters without changing the filters for the surge channel. This is also true for the filters for the roll channel.

When the yaw channel was tested, the output of the NASA adaptive algorithm and the optimal algorithm were different. If the nonlinear gains for the two algorithms are adjusted so that the two algorithms drive the simulator to the same amount of actuator extensions, the simulator angular velocity output of the NASA adaptive algorithm will have higher magnitude but be sustained for shorter time than the output of the optimal algorithm. It is not clear which output will result in better simulation effects.

It was found that in most input cases, to generate the specific force at the simulator pilot's head or the simulator angular velocity with the same magnitude, the NASA adaptive algorithm and the optimal algorithm would result in about the same amount of actuator extensions. The advantages of the optimal algorithm over the NASA adaptive algorithm are that the optimal algorithm eliminated the negative motion cues, generated outputs with better shapes in many simulation cases, and did not lose some desirable motion cues while the NASA adaptive algorithm did in some input cases.

REFERENCES

1. Gundry, A. J.; Thresholds to Roll Motion in a Flight Simulator, AIAA Visual and Motion Simulation Conference, Dayton, Ohio, April, 1976.
2. Clark, B., and Stewart, J. D.; Effects of Angular acceleration on Man: Thresholds for the Perception of Rotation and the Oculogyral Illusion, *Aerospace Medicine*, September, 1969.
3. Reid, L. D. and Nahon, M. A.; Flight Simulation Motion-base Drive Algorithms: Part 1 - Developing and Testing the Equations, UTIAS Report No. 296, CN ISSN 0082-5255, December 1985.
4. Reid, L. D. and Nahon, M. A.; Flight Simulation Motion-base Drive Algorithms: Part 2 - Selecting the System Parameters, UTIAS Report No. 307, CN ISSN 0082-5255, May 1986.
5. Martin, D. J., Jr.; *A Digital Program for Motion Washout on Langley's Six-degree-of-freedom Motion Simulator*, NASA Contractor Report 145219 Contract NAS1-14600, July 1977.
6. Sivan, R., Ish-shalom, J. and Huang J. K.; An Optimal Control Approach to the Design of Moving Flight Simulators, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-12, no.6, November/December 1982, pp. 818-827.
7. Conrad, B., Schmidt, S. F., and Douvillier, J. G.; *Washout Circuit Design for Multi-degrees-of-freedom Moving Base Simulators*; AIAA Paper No. 73-929, AIAA Visual and Motion Simulation Conference, September 1973.
8. Ogata, Katsuhiko, *Modern Control Engineering (Second Edition)*, N.J.: Prentice Hall, 1990.
9. Goldberg, J. M. and Fernandez, C., Physiology of Peripheral Neurons Innervating Semicircular Canals of the Squirrel Monkey. I, II, III, *Journal of Neurophysiology*, vol. 34, no. 4, 1971, pp. 635-685.
10. Fernandez, C., and Goldberg, J. M., Physiology of Peripheral Neurons Innervating Otolith Organs of the Squirrel Monkey. I, II, and III, *Journal of Neurophysiology*, vol. 39, no. 5, 1976, pp. 971-1009.
11. Ghista, D.N., Schmid, R., Buizza, A. and Zambarbieri, D., *Applied Physiological Mechanics*, vol. 1, OPA, Amsterdam, B.V., 1979, pp. 779-893.

12. Hosman, R. J. A. W., and van der Haart, J. C.; Vestibular Models and Thresholds of Motion Perception, Results of Tests in a Flight Simulator, Delft University of Technology, Department of Aerospace Engineering, Delft, The Netherlands, April 1978.
13. Benson, A. J., Spencer, M. B., and Stott, J. R. R.; Thresholds for the Detection of the Direction of Whole-Body, Linear Movement in the Horizontal Plane, Royal Airforce Institute of Aviation Medicine, Farnborough, Hampshire, United Kingdom. Reprint and Copyright by Aerospace Medical Association, Washington, DC, 1986.
14. Zacharias, G. L., Motion Cue Models for Pilot-Vehicle Analysis, Defense Documentation Center, Defense Logistics Agency, Cameron Station, Alexandria, VA, May, 1978.
15. Young, L. R. and Oman, C. M.; Model for Vestibular Adaptation to Horizontal Rotation, Aerospace Medicine, vol. 40, no. 10, 1969, pp. 1076-1080.
16. Young, L. R.; Perception of the Body in Space: Mechanisms, MIT, Cambridge, MA, 1982.
17. Hosman, R. J. A. W., and van der Haart, J. C.; Thresholds of Motion Perception and Parameters of Vestibular Models Obtained from Tests in a Motion Simulator. Effects of Vestibular and Visual Motion Perception on Task Performance, Delft University of Technology, Department of Aerospace Engineering, Delft, The Netherlands, April 1978.

Appendix A.

Vestibular Rotation Sensation Model Construction and Analysis

1. Introduction

The purpose of this study is to find a vestibular semicircular rotation sensation model that is most consistent with experimental results and can be justified by theoretical analysis, thus providing a reliable approximation to the rotation sensation function. This study is based on the reports presented by the many researchers who worked on the semicircular canal and rotation sensation function analysis. The model is first constructed step by step. The model parameters are then determined and then the model is carefully justified.

2. Model Construction

The vestibular rotation sensation model has been well studied by several authors. Steinhausen [A1] first developed a linear second order model of canal dynamics to explain the observed characteristics of vestibular induced eye movements in fish (pike) in 1931. This model was further refined by the "torsion-pendulum" model of Van Egmond, et al. [A2] in 1949, and is later developed from a systems approach by Mayne [A3]. The differential equation for this model is

$$I\ddot{\theta}_e + c\dot{\theta}_e + k\theta_e = I\alpha \quad (A-1)$$

where

I = Moment of inertia of the endolymph

c = Moment of viscous damping of the endolymph

k = Moment of elastic restoring force of the endolymph

θ_e = Angular displacement of the endolymph with respect to the head

α = Angular acceleration of the head with respect to an inertial axis

The following transfer function is obtained from Equation (A-1):

$$\frac{\theta_e(s)}{\alpha(s)} = \frac{1}{s^2 + \frac{c}{I}s + \frac{k}{I}} = \frac{\frac{I}{k}}{\frac{I}{k}s^2 + \frac{c}{k}s + 1} \quad (A-2)$$

For an overdamped system, Equation (A-2) can be written as

$$\frac{\theta_e(s)}{\alpha(s)} = \frac{\frac{I}{k}}{(1 + \tau_1 s)(1 + \tau_2 s)} \quad (A-3)$$

where $\tau_1 = \frac{c}{k}$, $\tau_2 = \frac{I}{c}$, and $\tau_1 > \tau_2$. Schmid, et al. [A4] show that cupula deflection ϕ_c

and the endolymph angular displacement θ_e are related by

$$\phi_c = -2 \frac{R A_c}{h A_{cp}} = -a \theta_e \quad (A-4)$$

where R = Central radius of the canal

A_c = Cross-sectional area of the canal

A_p = Cross-sectional area of the cupula

h = Height of the cupula

and a is a nondimensional number. Substituting (A-4) into (A-3) results in the transfer function relating cupula angular deflection to an angular acceleration input:

$$\frac{\phi_c(s)}{\alpha(s)} = \frac{a \tau_1 \tau_2}{(1 + \tau_1 s)(1 + \tau_2 s)} \quad (A-5)$$

The numerator in Equation (A-5) is equal to the K in the numerator in the transfer function given by Zacharias [A5].

Further studies showed that a complete vestibular rotational perception model is more complex than the torsion-pendulum model. Young and Oman [A6] formulated an

adaptation operator and cascaded it with the torsion-pendulum model to resolve the conflicts between the sensed response predicted by the torsion-pendulum model and the perceptual response measured in experiments. The addition of adaptation results in the following transfer function:

$$\frac{\phi_c(s)}{\alpha(s)} = K \cdot \frac{\tau_a s}{1 + \tau_a s} \cdot \frac{1}{(1 + \tau_1 s)(1 + \tau_2 s)} \quad (\text{A-6})$$

Zacharias [A5] noted the following in regard to an additional lead component of the form $(1 + \tau_L s)$:

"In addition to the adaptation just discussed, there appears to be evidence of lead sensitivity in vestibular processing of angular velocity information. In studying postural reactions to induced body tilt in 1970, Nashner found it necessary to augment the torsion pendulum model with a lead term having a 17 msec time constant, in order to fit reflex latencies to large amplitude disturbances. As noted by Ormsby, this type of lead behavior is not inconsistent with the vestibular nystagmus frequency responses reported by Benson, in which a high frequency gain rise was noted, consistent with a lead operator having a 60 msec time constant. Finally, in their investigation of primary afferent response of squirrel monkeys to rotational stimuli, Fernandez and Goldberg found that the population average frequency response could be best fit with the inclusion of a lead term having a 50 msec time constant."

A model representing both the semicircular canal and the peripheral neuron transduction dynamics is now established:

$$\frac{\phi_c(s)}{\alpha(s)} = K \cdot \frac{\tau_a s}{1 + \tau_a s} \cdot \frac{1 + \tau_L s}{(1 + \tau_1 s)(1 + \tau_2 s)} \quad (\text{A-7})$$

Fernandez and Goldberg [A7] determined the vestibular parameters for the squirrel monkey by direct measurement of the afferent nerves due to various angular acceleration inputs of different amplitudes and frequencies. Their transfer function relates the afferent firing rate of the vestibular nerve to the angular acceleration input:

$$\frac{\text{AFR}(s)}{\alpha(s)} = k \cdot \frac{\tau_a s}{1 + \tau_a s} \cdot \frac{1 + \tau_L s}{(1 + \tau_1 s)(1 + \tau_2 s)} \quad (\text{A-8})$$

where the sensitivity k is different than the sensitivity K given in Equation (A-7).

3. Parameter Determination

Fernandez and Goldberg [A7] determined the average transfer function parameters for the squirrel monkey as

$$\frac{AFR(s)}{\alpha(s)} = k \cdot \frac{80s}{1+80s} \cdot \frac{1+0.049s}{(1+5.7s)(1+0.003s)} \quad (A-9)$$

where the sensitivity k was estimated as $3.44 \text{ spikes sec}^{-1}/\text{deg sec}^{-2}$ for a constant acceleration input. Parameters for man are more difficult to measure because direct detection of the afferent nerve outputs of the vestibular system cannot be done and therefore most experiments were based on subjective responses or nystagmus tests.

The objectives of several researchers were to determine the values of the parameters based on the subjective response of humans. Van Egmond, et al. [A2] reported that τ_1 and τ_2 had values of about 10 seconds and 0.1 seconds respectively for man in 1949. The values were based on the verbal response of highly trained subjects subjected to various motion inputs in both a rotating chair and a torsion swing. Meiry (from Zacharias [A5]) measured detection latency as a function of angular acceleration step size, and found a long time constant of 7 seconds for roll-axis rotation about the earth-vertical axis. Guedry (from Zacharias [A5]) used a short period rotational stimulus consisting of an acceleration pulse doublet, and a response measure of apparent displacement, and found values of 16 seconds and 7 seconds in yaw and pitch, respectively about an earth-vertical axis. Malcolm and Melvill Jones (from Zacharias [A5]) investigated the response to earth-vertical rotation about all three body axes by using a velocity step as the stimulus, and measured the elapsed time to zero perceptual response. They also measured the slow phase velocity (SPV) nystagmus of vestibular

induced compensatory eye movements, thus providing two separate measures of canal function. They were able to derive two sets of long time constants for the three axes with the torsion pendulum model driving both perceived angular velocity and eye velocity. Their results are summarized in Table A.1.

Table A.1. Torsion Pendulum Long Time Constants (from Zacharias [A5]).

	Yaw	Pitch	Roll
subjective sensation	10.2 ± 1.8	5.3 ± 0.7	6.1 ± 1.2
nystagmus SPV	15.6 ± 1.2	6.6 ± 0.7	4.0 ± 0.4

It should be noted that all the above estimations were based on the model expressed as the transfer function in Equation (A-5).

Young and Oman [A6] later fit these measured parameters to a more complex model. Young and Oman's model also resolved the apparent inconsistency between the nystagmus and subjective response measures for yaw rotation. Figure A.1 shows Young and Oman's model. The torsion-pendulum model of the canals drives both perceptual and nystagmus response channels. Each channel has its own adaptation time constant. A single long time constant of 16 seconds is used for the torsion-pendulum model. A combination of the torsion-pendulum model and the adaptation mechanism generated a different response to a step input for each channel. When the model is stimulated by step inputs, the apparent time constant of the nystagmus decay is 16 seconds, whereas the apparent time constant for sensation decay is 10 seconds. This model predicts a result for earth-vertical yaw rotation consistent with the data given in Table A.1, but was never extended for roll and pitch rotation.

It can be inferred that the long time constant τ_1 measured by Van Egmond, et al. [A2], Meiry and Malcolm and Melvill-Jones (from Zacharias [A5]) does not actually

represent the semicircular canal parameter in the model, but is an overall dynamics parameter representing the decay speed of the rotational sensation to a step angular velocity input. Zacharias [A5] suggests each axis of rotation has an equivalent “body axis” canal pair with a distinct time constant. These “body axis” time constants can then be transformed into the three physical canal pairs. The psychophysical results show each of the three canal pairs having a distinct value for τ_1 , especially for yaw. However, physiological results based on afferent responses by Fernandez and Goldberg [A7] show the same value for τ_1 for all three canal pairs. Zacharias [A5] suggests the differences shown in the psychophysical results may occur at a central origin at the perceptual level.

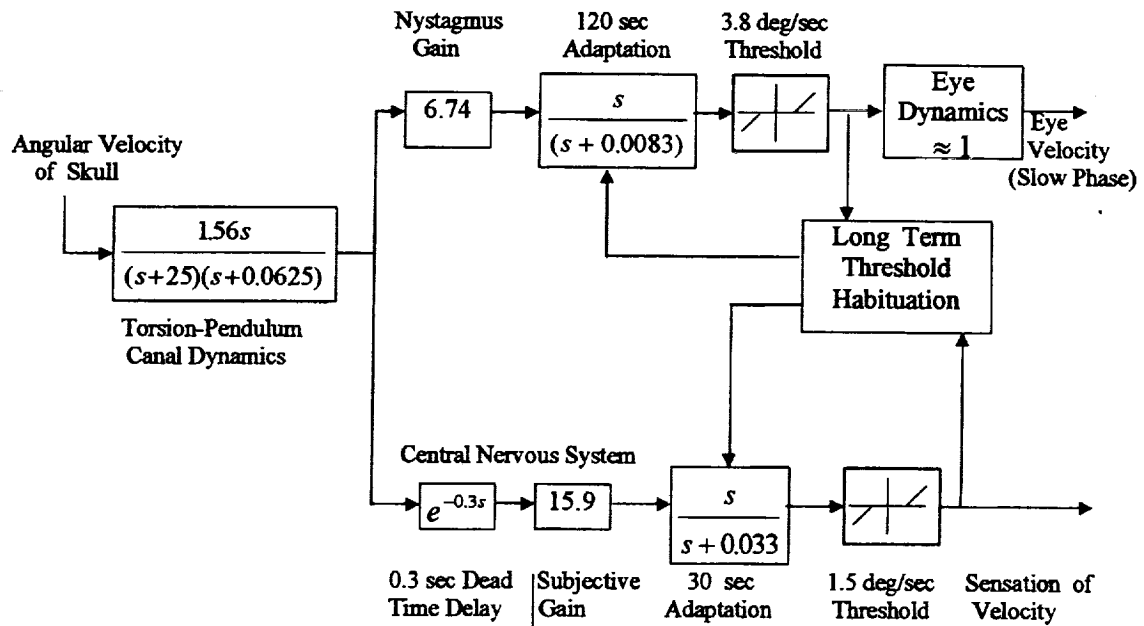


Figure A.1. Adaptation Model for Earth-Vertical Rotation (from [A3]).

Fernandez and Goldberg [A7] made the following observation in regard to the short time constant τ_2 :

"Our observations, since they extended only to 8.0 Hz, are insufficient to provide a direct experimental measure of τ_2 . Some estimate of this constant can, however, be made from hydrodynamic considerations. What seems to be required is a solution of the Navier-Stokes equation for the complicated geometry represented by the canal, the associated ampulla, and the utricle. No one to our knowledge has accomplished this, though Steer has solved the equation for a straight tube in 1967. The approximate value of the time constant, so derived, is $\tau_2 = (\rho r^2 / \eta \beta^2)$, where β is the first zero of the zeroth-order Bessel function of the first kind; ρ and η are, respectively, the density and viscosity of the endolymph, and r is the internal radius of the tube, in this case the radius of the membranous canal. This radius has been measured by Igarashi and leads to a value of τ_2 equal to 0.005 seconds in man and 0.003 seconds in the squirrel monkey."

Several authors did experiments to determine the value of τ_L . Their findings are summarized in Table A.2.

Table A.2. Lead Sensitivity to Rotational Stimuli (From Zacharias [A5]).

τ_L (sec)	Measure	Source
0.017 (man)	posture control	Nashner
0.06 (man)	nystagmus	Benson and Ormsby
0.05 (squirrel monkey)	primary afferent	Fernandez and Goldberg

From experiments with human subjects, Van Egmond, et al. [A2] showed that the perceived angular velocity $\hat{\omega}$ is proportional to the cupula deflection ϕ_c by the long time constant τ_1 , resulting in the transfer function

$$\frac{\hat{\omega}(s)}{\omega(s)} = \frac{\tau_1 s}{(1 + \tau_1 s)(1 + \tau_2 s)} \quad (\text{A-10})$$

As shown in Equation (A-9), Fernandez and Goldberg [A7] show a gain sensitivity k between the input stimulus and the afferent firing rate that was estimated at 3.44 spikes $\text{sec}^{-1} / \text{deg sec}^{-2}$. Ormsby (from Zacharias [A5]) proposed that the perceived angular velocity $\hat{\omega}$ is proportional to the afferent firing rate. While no one to date has experimentally obtained this parameter, Zacharias [A5] noted that Curry, et al. provided an estimate of the overall gain between perceived and input angular velocity based on angular acceleration thresholds.

From subjective pilot measurements of angular acceleration thresholds on a moving base platform, Hosman and van der Vaart [A8] show the perceived input threshold a_{\min} as

$$a_{\min} = \frac{b_{\min}}{|H(\omega)|} \quad (\text{A-11})$$

where b_{\min} = Minimum sensed amplitude of the afferent firing rate

$|H(\omega)|$ = Modulus of the vestibular transfer function

From the measured frequency response of a_{\min} the following transfer function is obtained, neglecting gain sensitivity and adaptation:

$$H(s) = \frac{1 + 0.1097 s}{(1 + 5.924 s)(1 + 0.005 s)} \quad (\text{A-12})$$

These results are based upon roll and pitch acceleration thresholds; yaw thresholds were not measured. The value for τ_1 agrees well with the value obtained by Fernandez and Goldberg. The value obtained for τ_L is nearly twice the nystagmus value obtained by Benson and Ormsby as given in Table A.2.

From the results reported by the authors mentioned in this section, a transfer function that can best describe the vestibular rotational sensation system is proposed:

$$\frac{\text{AFR}(s)}{\alpha(s)} = 3.44 \frac{80s}{1 + 80s} \cdot \frac{1 + 0.06s}{(1 + 5.73s)(1 + 0.005s)} \quad (\text{A-13})$$

It should be noted that the parameters in the above transfer function are by no means exact, but are of the correct order of magnitude, thus making the transfer function a meaningful approximation to the real dynamics of the vestibular rotation sensation system. The frequency response of the transfer function given in Equation (A-13) with

gain $k = 1$ is shown in Figure A.2. Both the torsion-pendulum model and the complete sensation model with lead and adaptation mechanisms included are both shown.

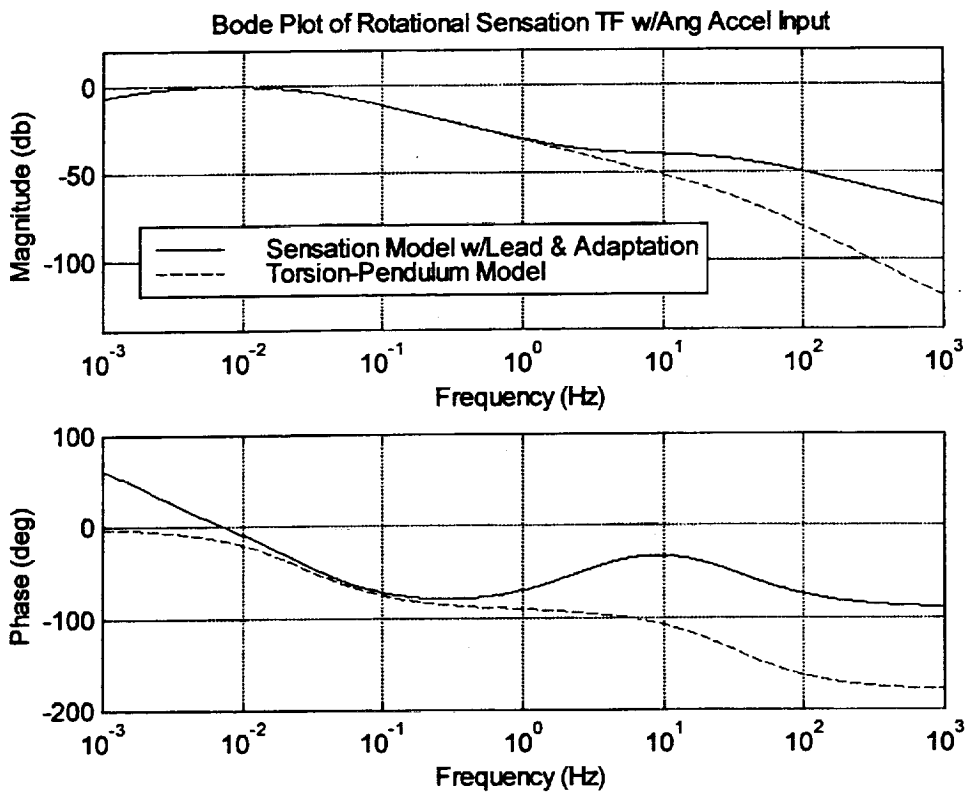


Figure A.2. Frequency Response of Vestibular Rotation Sensation System.

The sensory function of the semicircular canal can be described by observing the frequency response of the torsion-pendulum model. In the range of normal head movement from 0.05 to 5.0 Hz (Mayne, [A3]), the gain response decreases by 20 dB/decade with the phase close to minus 90 degrees. In this frequency range the canal functions as an “integrating accelerometer” or an angular velocity transducer. At very low frequencies less than 0.01 Hz, the phase approaches zero degrees, thus functioning as an accelerometer. At very high frequencies greater than 100 Hz, the phase approaches minus 180 degrees, thus functioning as an angular displacement transducer. The effects

of adaptation and lead on rotational sensation are apparent; adaptation influences low frequencies below 0.01 Hz while lead influences high frequencies greater than 10 Hz.

4. Physiological Interpretation

Modern theories of the operation of the semicircular canal receptors are based on the assumption that the nervous impulses are generated by deflection of the hairs in the sensory cells as a result of the cupula displacement. Input acceleration is first transformed into cupula deflection by the cupula-endolymph system. Then the cupula deflection is further transformed into electrical impulses by the mechano-neural transduction system consisting of sensory hair cells, afferent nerves and efferent nerves.

The cupula-endolymph system was likened to an overdamped linear torsion-pendulum by Steinhausen [A1] in 1931. A rigorous analytical evaluation of the dynamics of the endolymph motion in the semicircular canals was made by Steer [A9] in 1967. The torsion-pendulum model is well accepted by different authors without controversy upon the form of the model.

Besides the torsion-pendulum dynamics there are some additional terms in the complete transfer function in Equation (A-8). These terms could be grouped as

$$\frac{k\tau_a s(1+\tau_L s)}{(1+\tau_a s)} \quad (\text{A-14})$$

representing an adaptation-lead mechanism. Controversy occurred when different authors tried to interpret this adaptation-lead mechanism primarily because the precise mechanism of hair cell stimulation is not yet understood.

The first controversy is whether the adaptation arises in the mechanics of the cupula-endolymph system or in the mechano-neural system. Goldberg and Fernandez [A8] presented a good discussion of the origin of the adaptation:

"Two observations tend to suggest that the adaptation arises in the mechano-neural system. The first is the clear adaptation seen by Lowenstein when polarizing currents were applied to the vestibular nerve of the thornback ray. Presumably, the currents acted directly on the nerve terminals. The second is the fact that units differ greatly in their adaptive properties. Were adaptation of mechanical origin, one would have to assume that hair cells differ in the way they are mechanically coupled to the motion of the cupula. An assumption which appears to us more reasonable is that the adaptation reflects the physiology of the hair cells and/or of the afferent nerve terminals. Another, perhaps unlikely, possibility is that the adaptation results from the activation of the efferents innervating the sensory epithelium."

Controversy arose again about the origin of the lead operator. Since no mechanism was found in the cupula-endolymph system that could provide a reasonable interpretation for the lead operator, most authors suggested that the lead operator arose from the mechano-neural system. Fernandez and Goldberg [A7] suggested that the lead operator implies the sensory hair cells are sensitive to both the displacement and the velocity of the cupula. The time constant τ_L reflects the relative sensitivities to these two aspects of the cupular motion. They further suggested that the adaptation-lead mechanism represents the transfer function between the discharge frequency F and the cupula deflection ϕ :

$$\frac{F(s)}{\phi(s)} = \frac{k\tau_a s(1 + s\tau_L)}{1 + s\tau_a} \quad (\text{A-15})$$

This expression implies that both adaptation and lead arise from the mechano-neural system.

Schmid, Buizza, et al. [A4] gave two other plausible interpretations for the adaptation-lead mechanism. One is based on the observation that there are two types of sensory cells in the vestibular sensory epithelia of mammals. All the sensory cells are assumed to be characterized by a transfer function of the type

$$\frac{F_1(s)}{\phi(s)} = \frac{K * T^* s}{1 + T^* s} \quad (A-16)$$

where F_1 is the afferent nerve output, ϕ is the cupula deflection, K_1 is the sensitivity factor and T^* is the time constant of the transduction process. It is further assumed that the two different types of sensory cells work in parallel with a different T^* . Then the overall transfer function of the mechano-neural transduction system can be expressed as

$$\frac{F(s)}{\phi(s)} = \frac{K_{21} T_1 s}{1 + T_1 s} + \frac{K_{22} T_2 s}{1 + T_2 s} \quad (A-17)$$

By simple manipulation the following transfer function can be obtained:

$$\frac{F(s)}{\phi(s)} = \frac{K_{21} T_1 s (1 + T_3 s)}{(1 + T_1 s) (1 + T_2 s)} \quad (A-18)$$

where $T_1 \gg T_2$ and $T_3 \approx 2T_2$. If T_2 is assumed negligible, then the transfer function can be reduced to an adaptation-lead operator and give a good interpretation to the origin of the adaptation-lead mechanism. A very small T_2 implies that a group of sensory cells have a very small adaptation time constant. Experiments have shown that the adaptation time constants of the vestibular mechano-neural system of squirrel monkeys range approximately from 30 seconds to infinitely long. The fact that no group of sensory cells with a small adaptation time constant were found experimentally conflicts with the first interpretation presented by Schmid, et al. [A4].

The second interpretation presented by Schmid, et al. [A4] is based on the existence of efferent pathways descending from the vestibular nuclei to the sensory epithelium, possibly with an inhibitory function. These efferents may represent a negative feedback that modifies the characteristics of the mechano-neural transduction. The average dynamics of all sensory cells is assumed to be

$$\frac{F_i(s)}{\phi(s)} = \frac{K * T^* s}{1 + T^* s} \quad (A-19)$$

The dynamics of the inhibitory process is described by means of a first order system with a time constant T_i much smaller than T^* . The complete mechano-neural transduction system can be represented by the block diagram in Figure A.2.

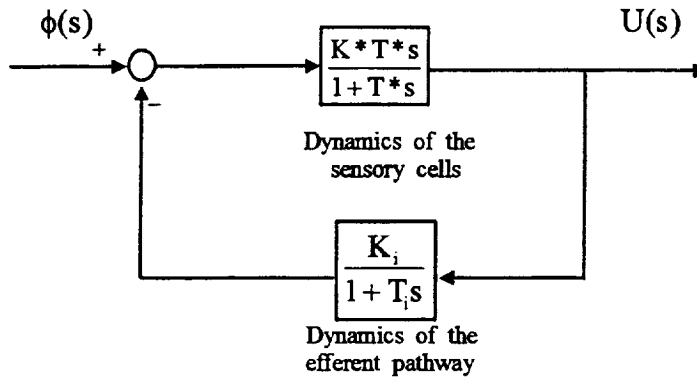


Figure A.2. A Closed-loop Interpretation of the Mechano-neural System.

The closed-loop transfer function can be obtained:

$$\frac{F(s)}{\phi(s)} = \frac{\frac{K * T^* s}{1 + T^* s}}{1 + \frac{K * T^* s}{1 + T^* s} \frac{K_i}{1 + T_i s}} = \frac{K * T^* s (1 + T_i s)}{1 + (K * K_i + 1) T^* s + T_i s + T^* T_i s^2} \quad (A-20)$$

By introducing the simplifications suggested by the condition $T_i \ll T^*$, it follows that

$$\frac{F(s)}{\phi(s)} = \frac{K * T^* s (1 + T_i s)}{\left[1 + (K * K_i + 1) T^* s\right] \left(1 + \frac{T_i}{K * K_i + 1} s\right)} \quad (A-21)$$

If $\frac{T_i}{K * K_i + 1}$ is small enough to be considered negligible, then the above transfer

function can be further reduced to the form of Equation (A-15).

4. Conclusion

The vestibular rotational sensation system has been well studied. Its model consists of a second order torsion-pendulum operator and an adaptation-lead operator. The torsion-pendulum operator arises from the cupula-endolymph system and functions as an angular velocity transducer in the range of normal head movement. The model is then augmented by the adaptation-lead operator that arises from the mechano-neural transduction system. The adaptation-lead operator most likely represents a closed-loop dynamic system consisting of sensory cells, afferent nerves as feed-forward path and efferent nerves as negative feedback path. Based upon the model parameters reported in the literature a transfer function best approximating the rotational sensation system is proposed.

5. References

- A1. Steinhausen, W., Observations of the Cupula in the Ampullae of the Semicircular Canals of the Labyrinth of a Living Pike, *Pflüger's Arch. ges Physiol.*, vol. 232, pp. 500-512, 1933, NASA Technical Translation NASA TT F-13, 665, May 1971.
- A2. Van Egmond, A. A. J., Groen, J. J., and Jongkees, L. B. W., The Mechanics of the Semi-circular Canal, *Journal of Physiology*, vol. 110, 1949, pp. 1-17.
- A3. Mayne, R., A Systems Concept of the Vestibular Organs, *Handbook of Sensory Physiology, Vestibular System*, vol. 6, part 3, pp. 493-560.
- A4. Schmid, R., Buizza, A. and Zambarbieri, D., *Applied Physiological Mechanics*, vol. 1, OPA, Amsterdam, B.V., 1979, pp. 779-893.
- A5. Zacharias, G. L., Motion Cue Analysis for Pilot-Vehicle Analysis, Department of Defense Report AMRL-TR-78-2, May 1978.
- A6. Young, L. R. and Oman, C. M.; Model for Vestibular Adaptation to Horizontal Rotation, *Aerospace Medicine*, vol. 40, no. 10, 1969, pp. 1076-1080.
- A7. Fernandez, C., and Goldberg, J. M., Physiology of Peripheral Neurons Innervating Semicircular Canals of the Squirrel Monkey. II. Response to Sinusoidal Stimulation and Dynamics of Peripheral Vestibular System. *Journal of Neurophysiology*, vol. 34, no. 4, 1971, pp. 661-675.
- A8. Hosman, R. J. A. W. and van der Vaart, J. C., Vestibular Models and Thresholds of Motion Perception. Results of Tests in a Flight Simulator. Delft University of Technology, Department of Aerospace Engineering, Report LR-265, April 1978.
- A9. Steer, R. W., The Influence of Angular and Linear Acceleration and Thermal Stimulation on the Human Semicircular Canal, Sc. D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, 1967.

Appendix B.

Phase 1 Rotational Output Figures

- Figure B.1. Classical Algorithm Roll Doublet Pulse Input.
1 m/s² peak, 10 second duration input.
- Figure B.2. Optimal Algorithm Roll Doublet Pulse Input.
1 m/s² peak, 10 second duration input.
- Figure B.3. NASA Adaptive Algorithm Roll Doublet Pulse Input.
1 m/s² peak, 10 second duration input.
- Figure B.4. UTIAS Adaptive Algorithm Roll Doublet Pulse Input.
1 m/s² peak, 10 second duration input.

CLASSICAL ROLL TEST CASE
DOUBLET INPUT

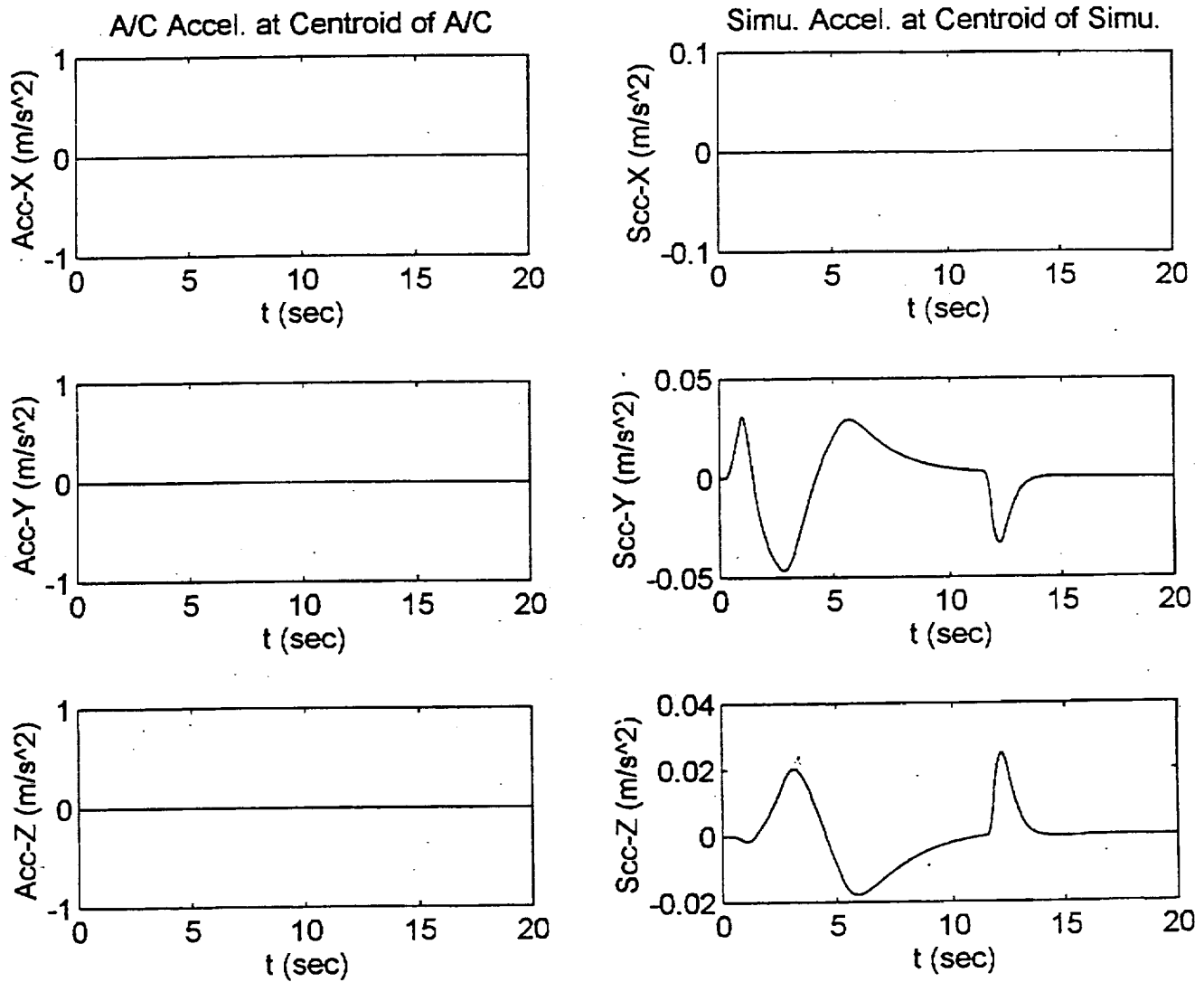


Figure B.1. Classical Algorithm Roll Doublet Pulse Input.

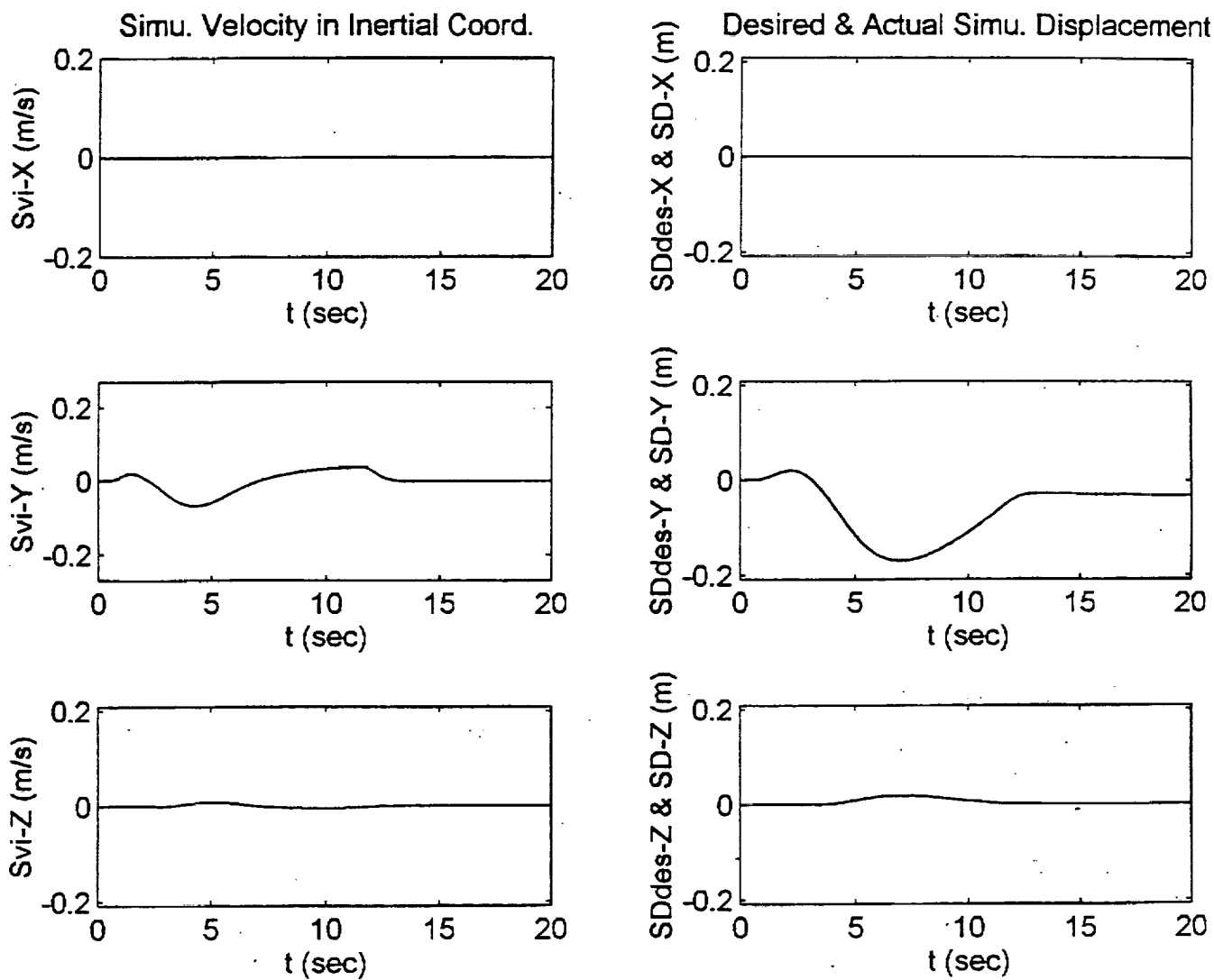


Figure B.1. Classical Algorithm Roll Doublet Pulse Input.

CLASSICAL ROLL TEST CASE
DOUBLET INPUT

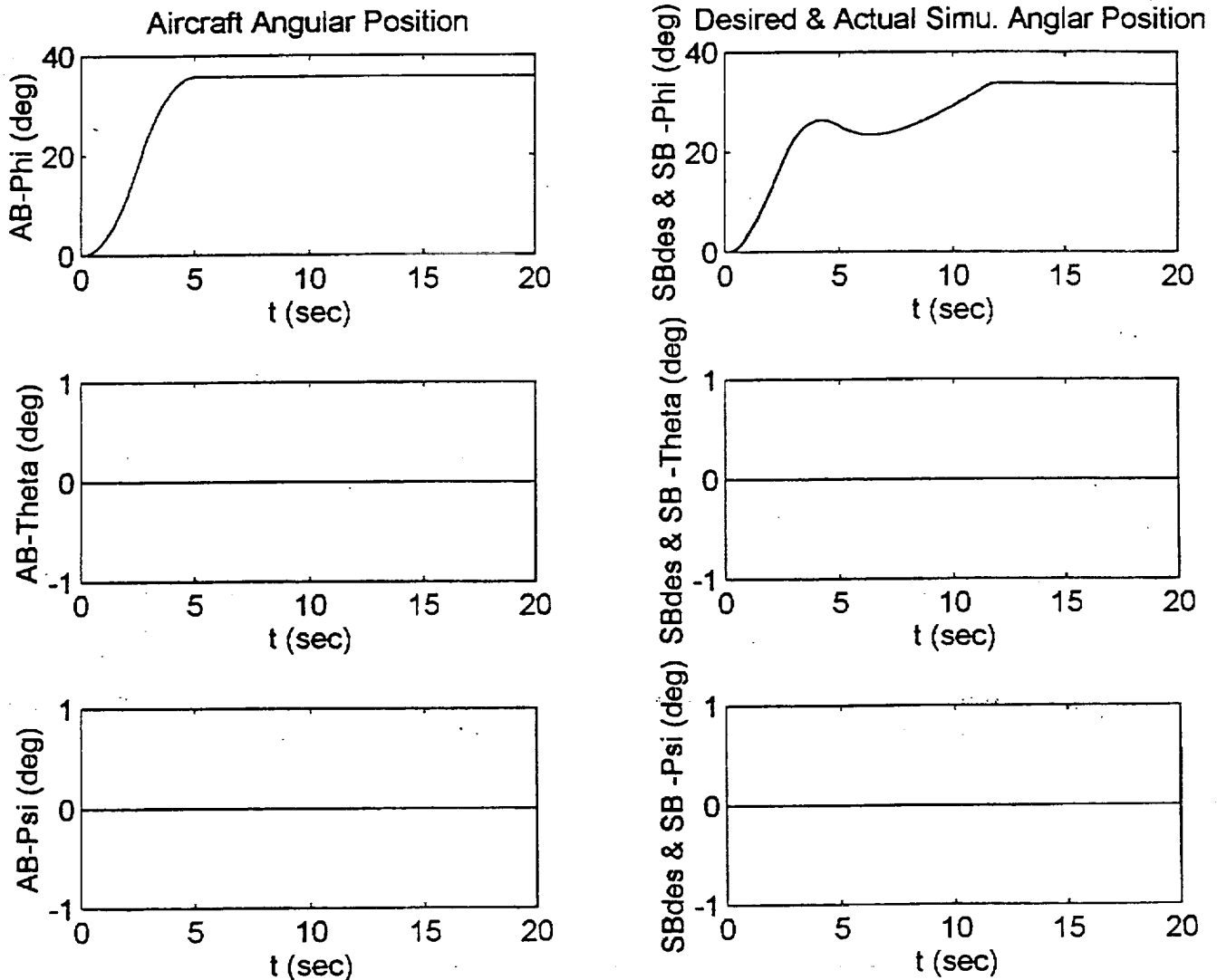


Figure B.1. Classical Algorithm Roll Doublet Pulse Input.

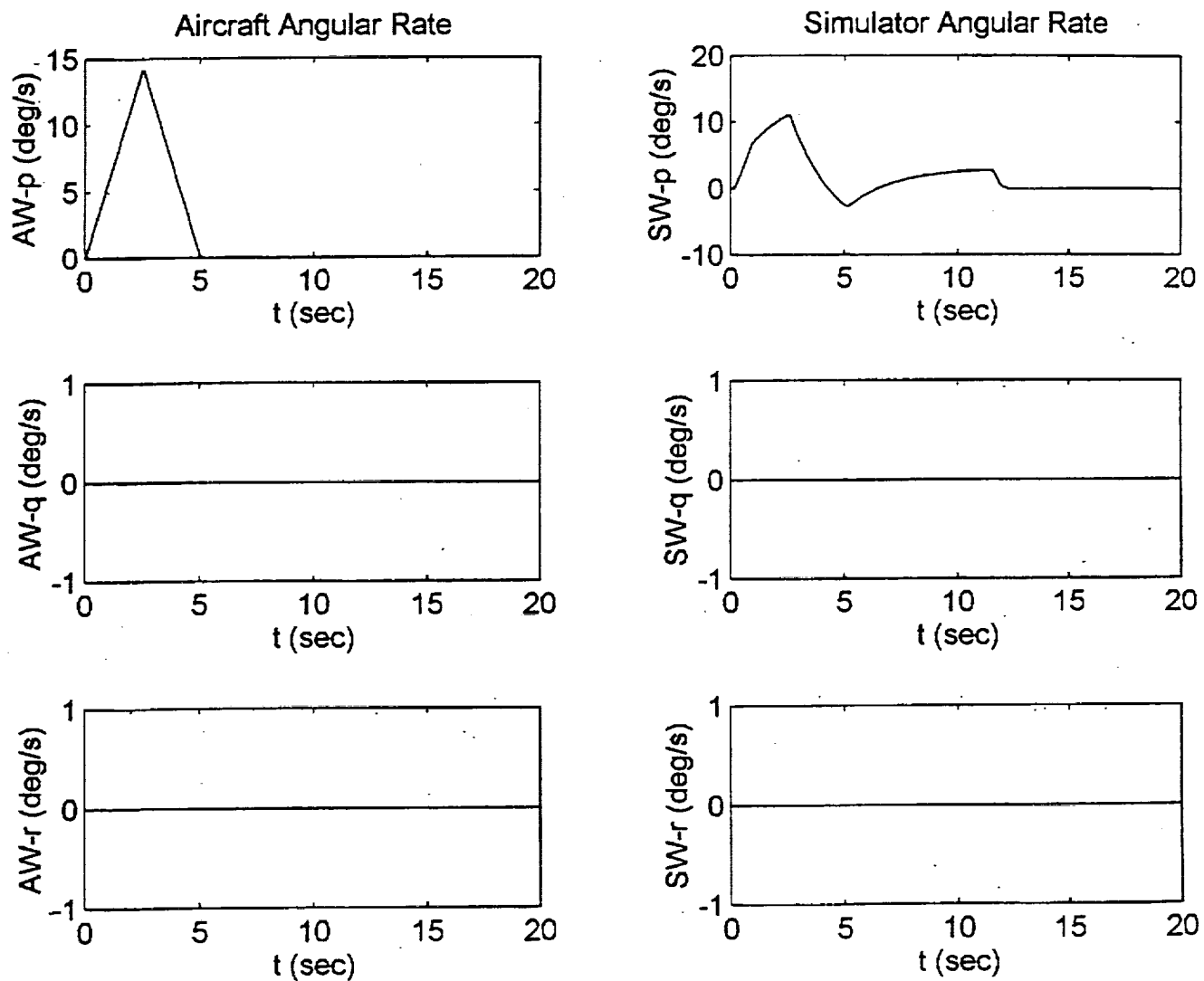


Figure B.1. Classical Algorithm Roll Doublet Pulse Input.

CLASSICAL ROLL TEST CASE
DOUBLET INPUT

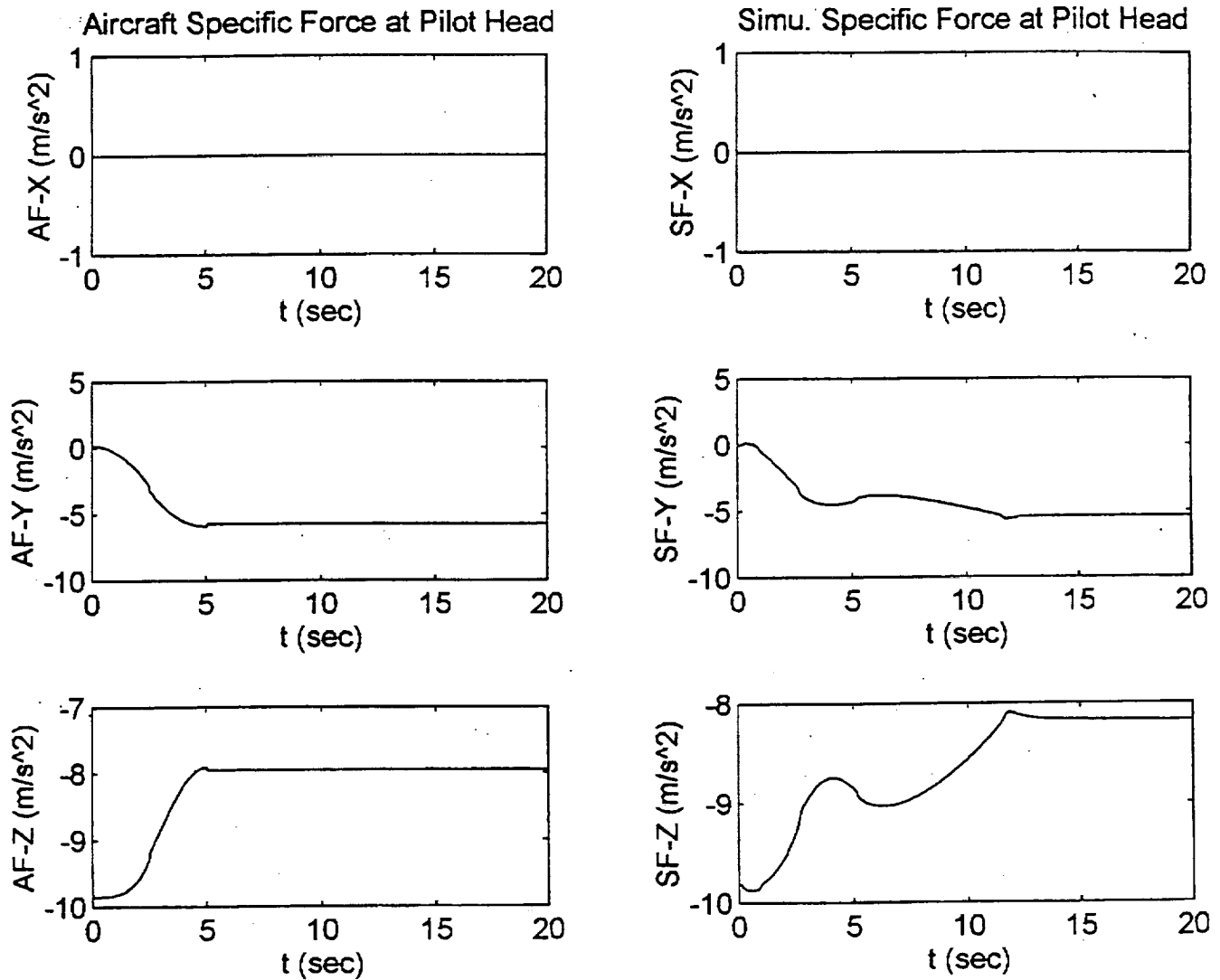


Figure B.1. Classical Algorithm Roll Doublet Pulse Input.

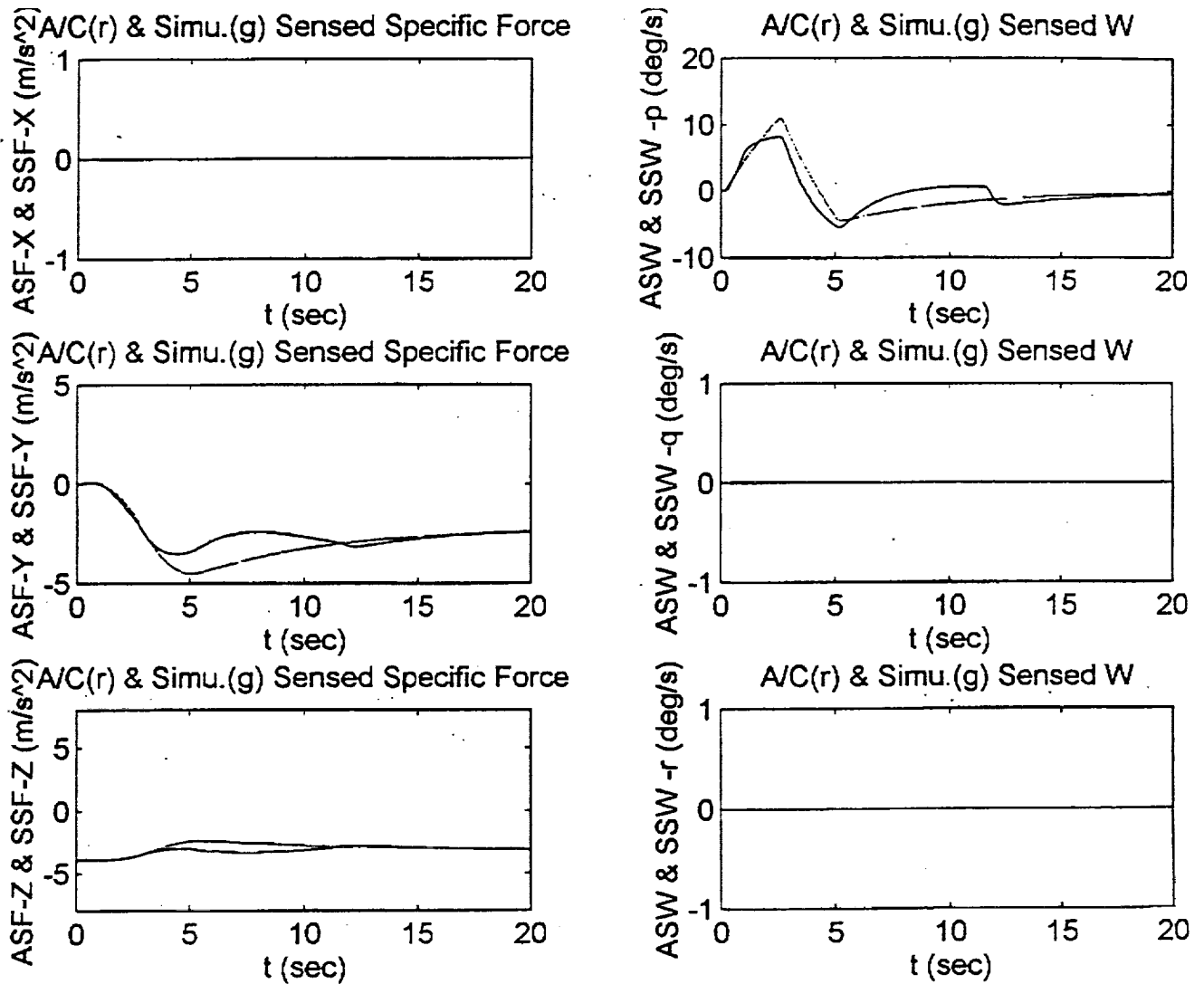


Figure B.1. Classical Algorithm Roll Doublet Pulse Input.

CLASSICAL ROLL TEST CASE
DOUBLET INPUT

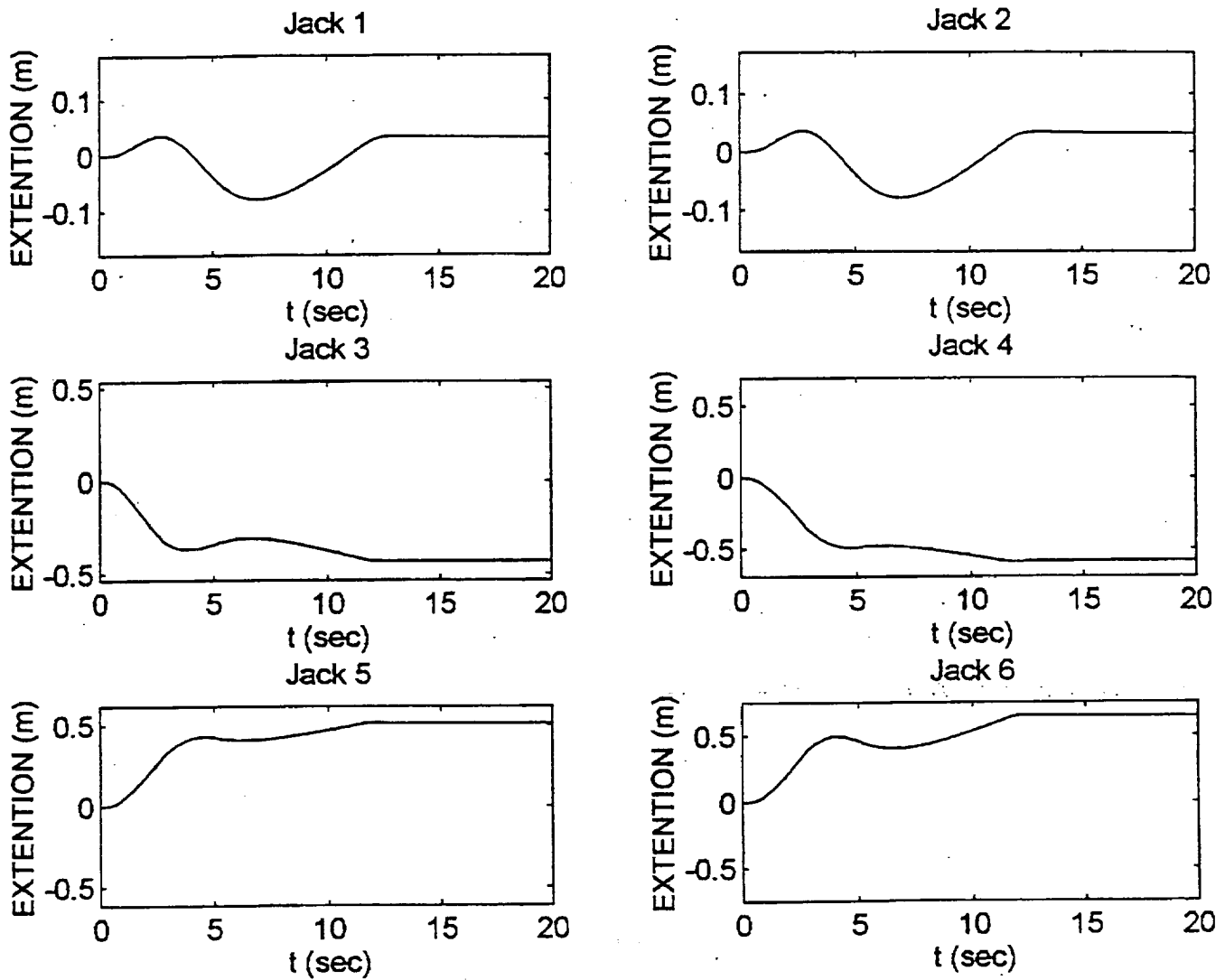


Figure B.1. Classical Algorithm Roll Doublet Pulse Input.

OPTIMAL ROLL TEST CASE
DOUBLET INPUT

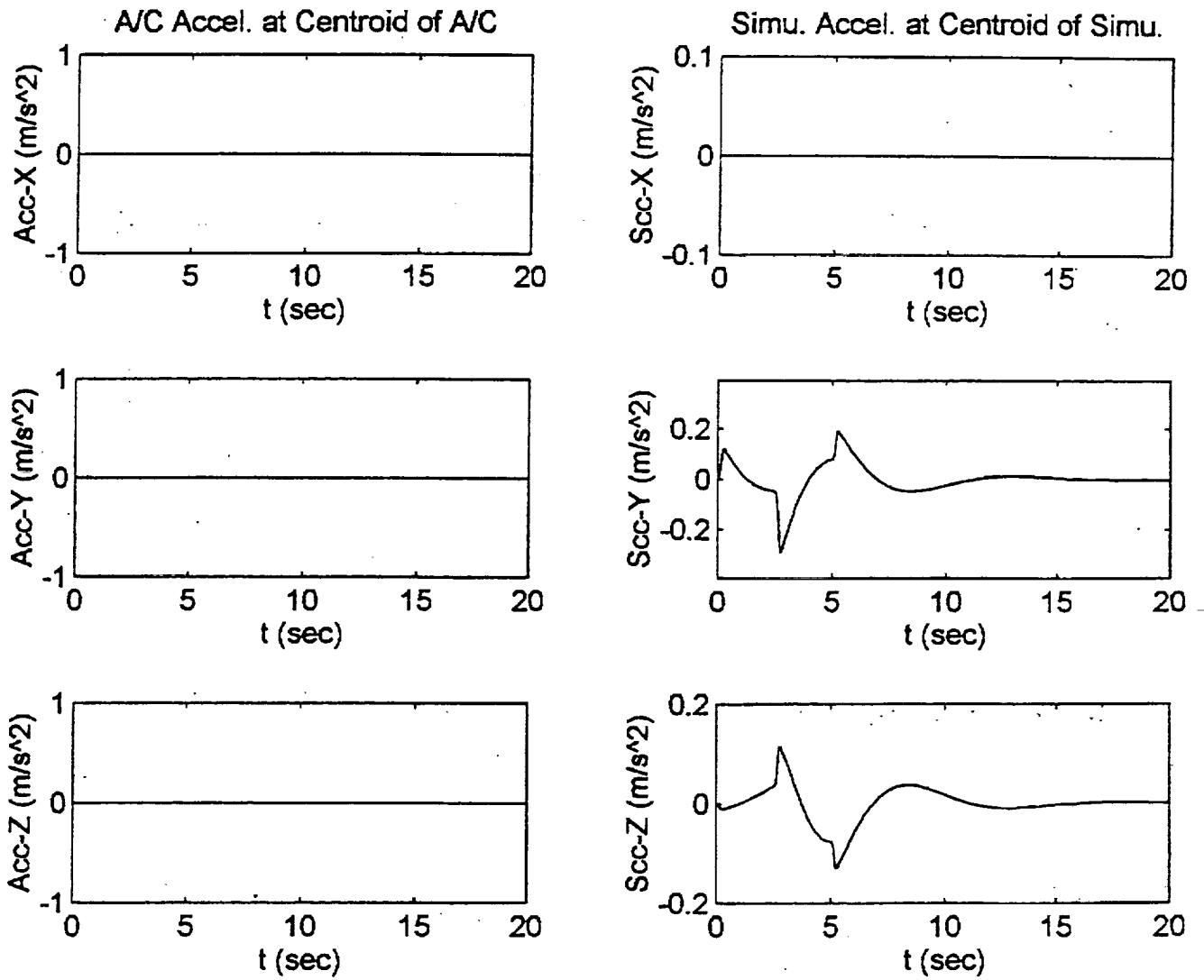


Figure B.2. Optimal Algorithm Roll Doublet Pulse Input.

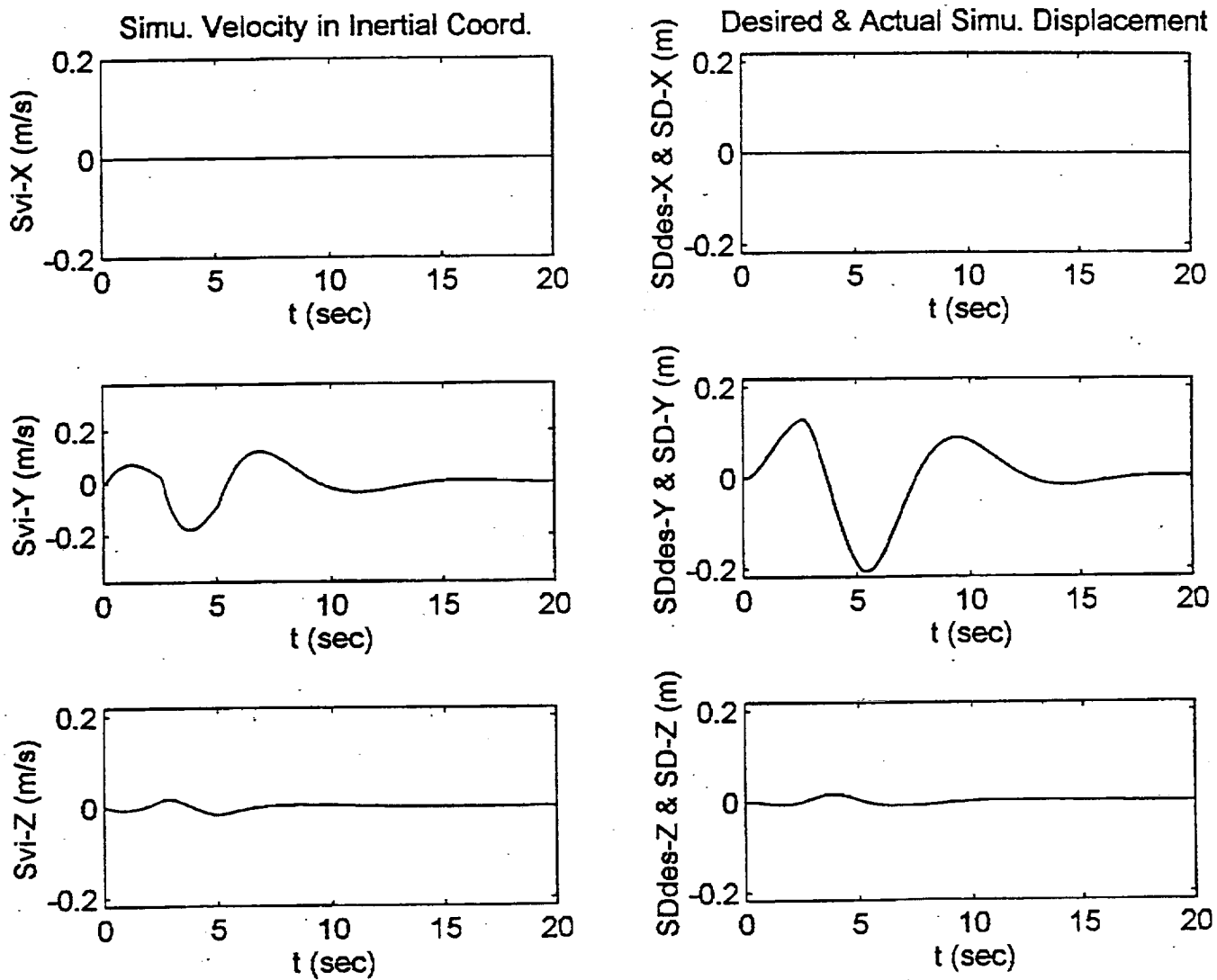


Figure B.2. Optimal Algorithm Roll Doublet Pulse Input.

OPTIMAL ROLL TEST CASE
DOUBLET INPUT

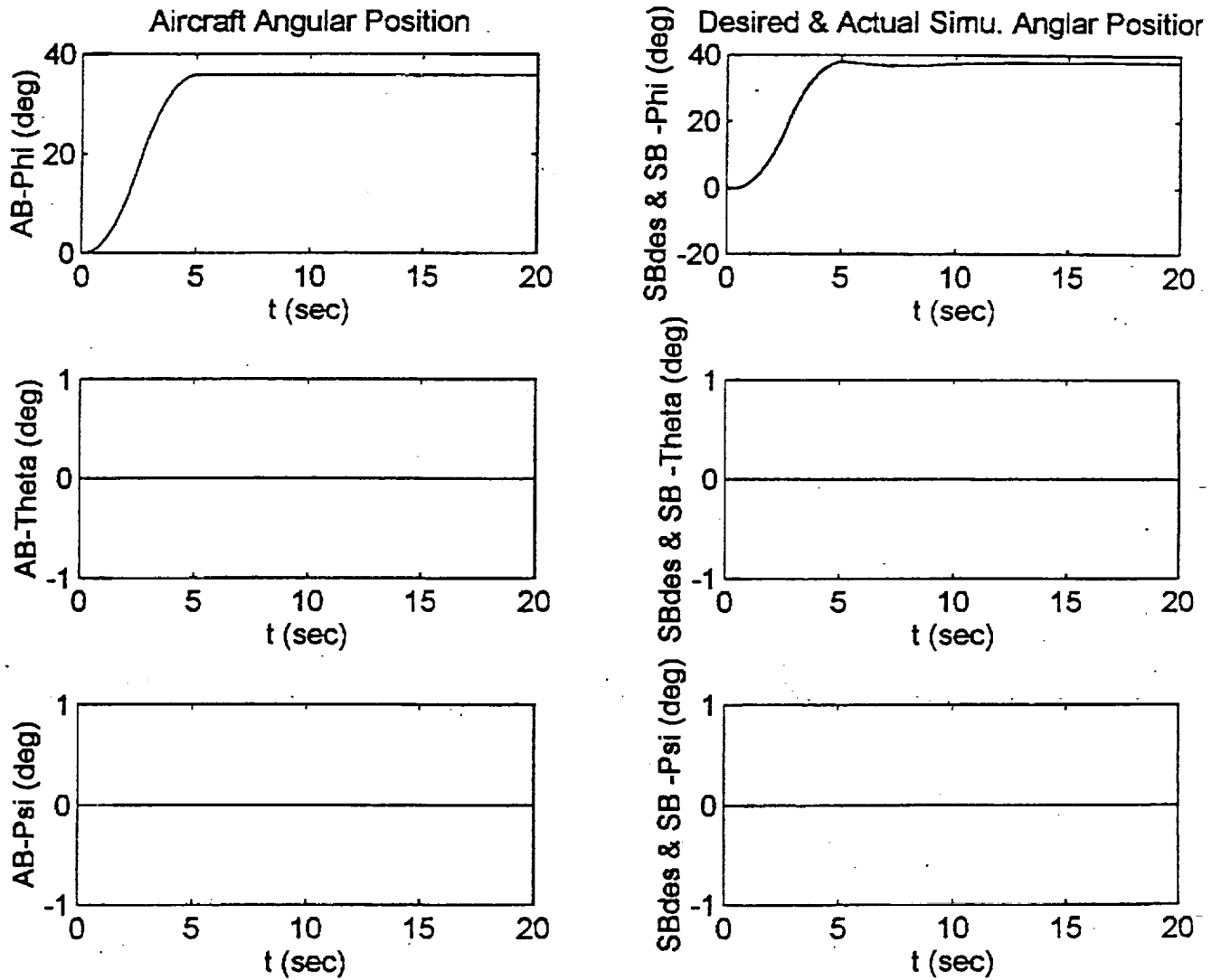


Figure B.2. Optimal Algorithm Roll Doublet Pulse Input.

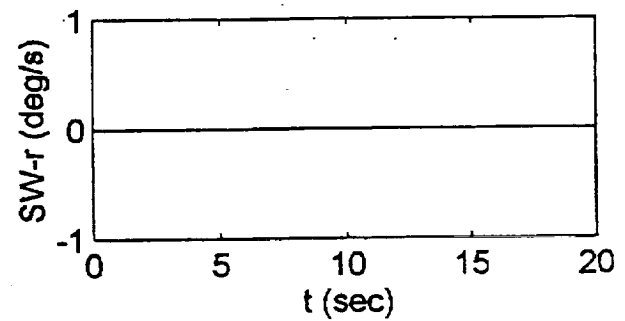
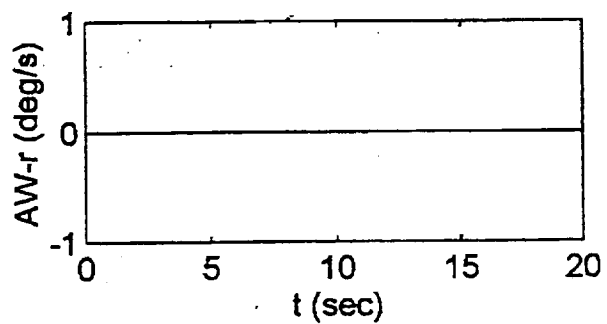
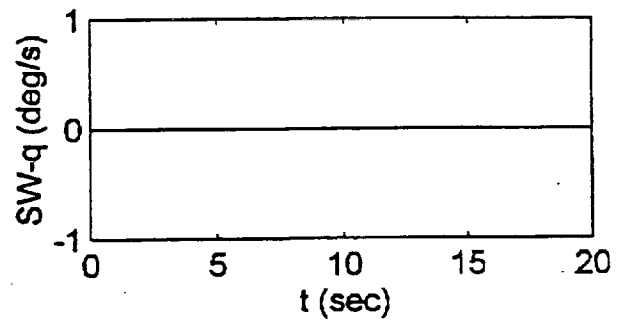
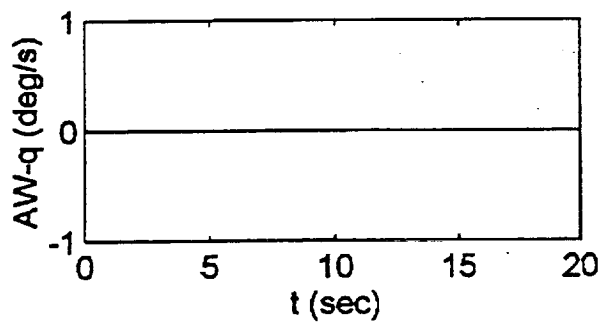
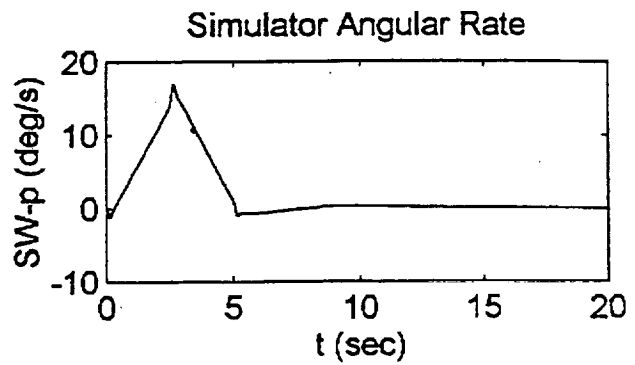
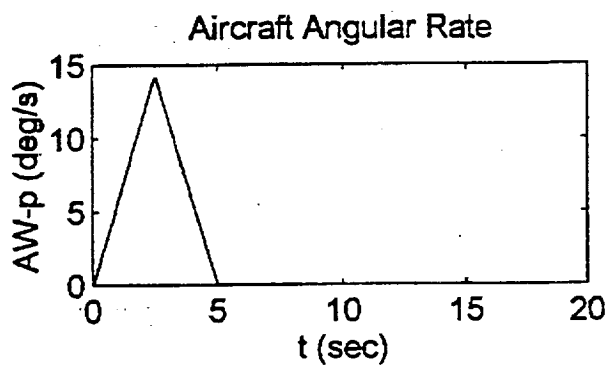


Figure B.2. Optimal Algorithm Roll Doublet Pulse Input.

OPTIMAL ROLL TEST CASE
DOUBLET INPUT

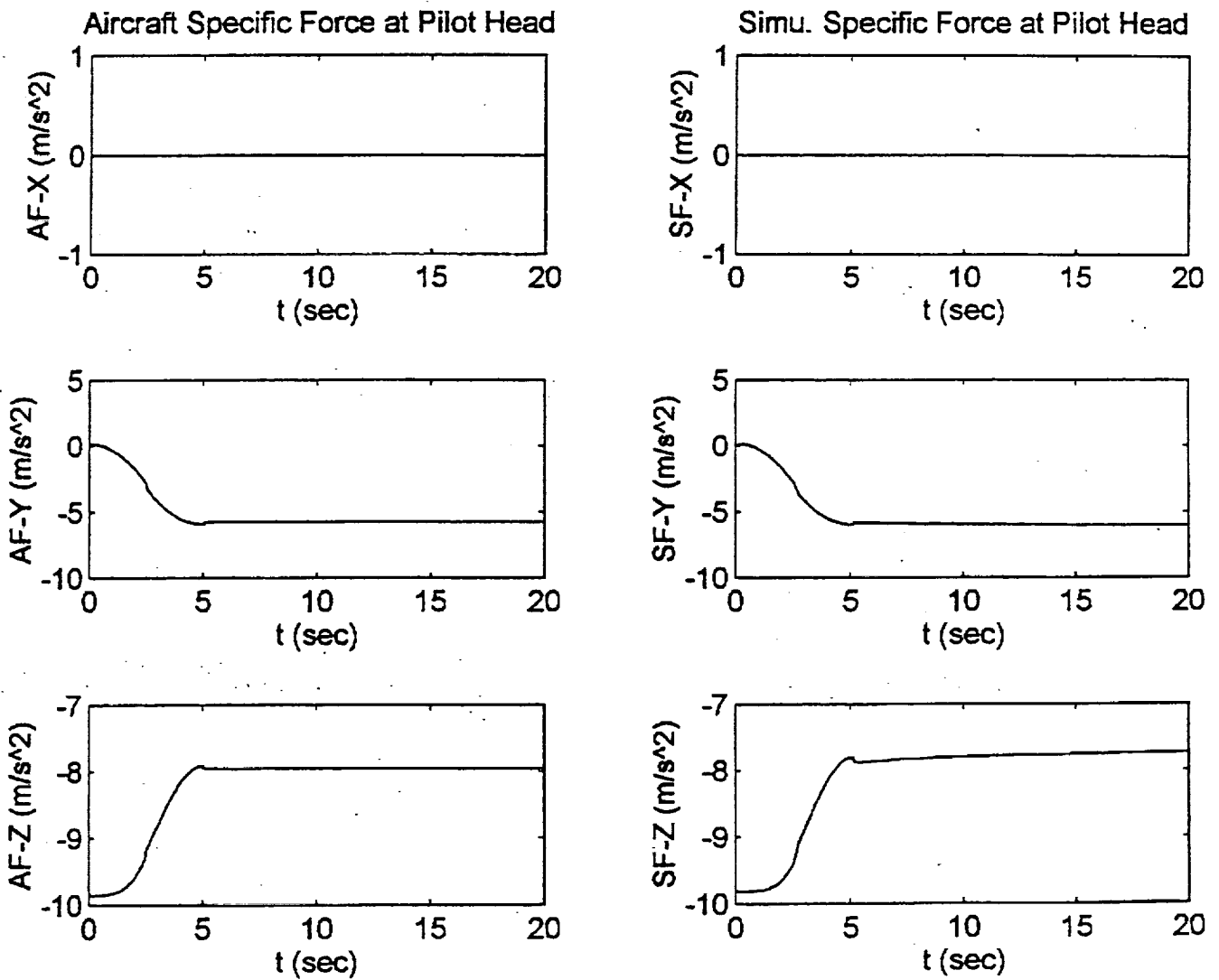


Figure B.2. Optimal Algorithm Roll Doublet Pulse Input.

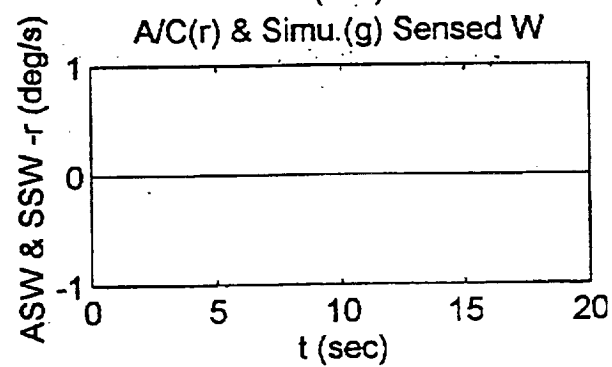
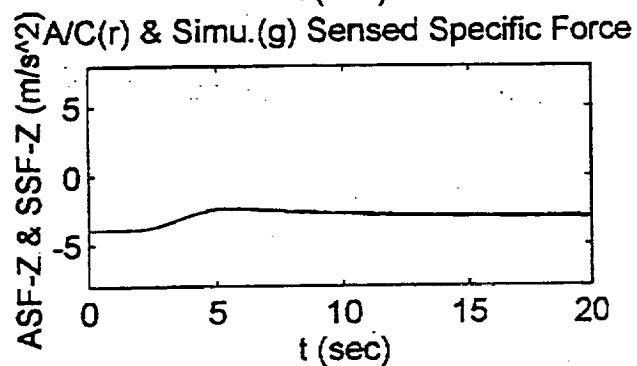
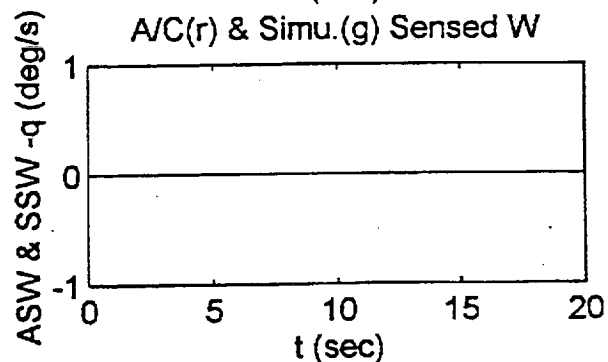
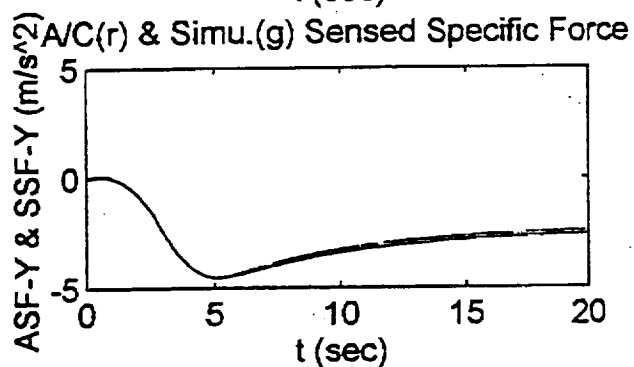
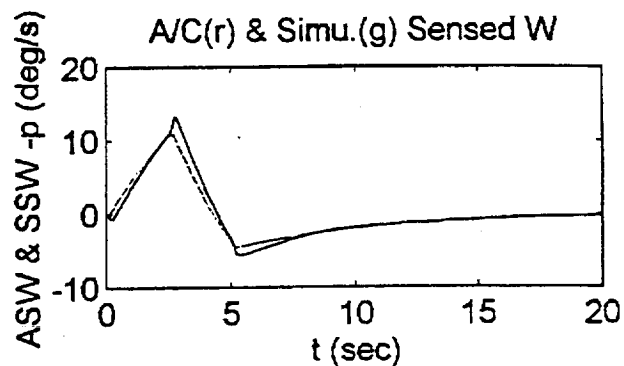
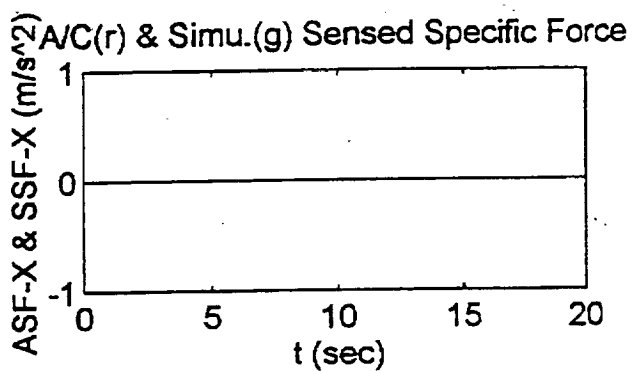


Figure B.2. Optimal Algorithm Roll Doublet Pulse Input.

OPTIMAL ROLL TEST CASE
DOUBLET INPUT

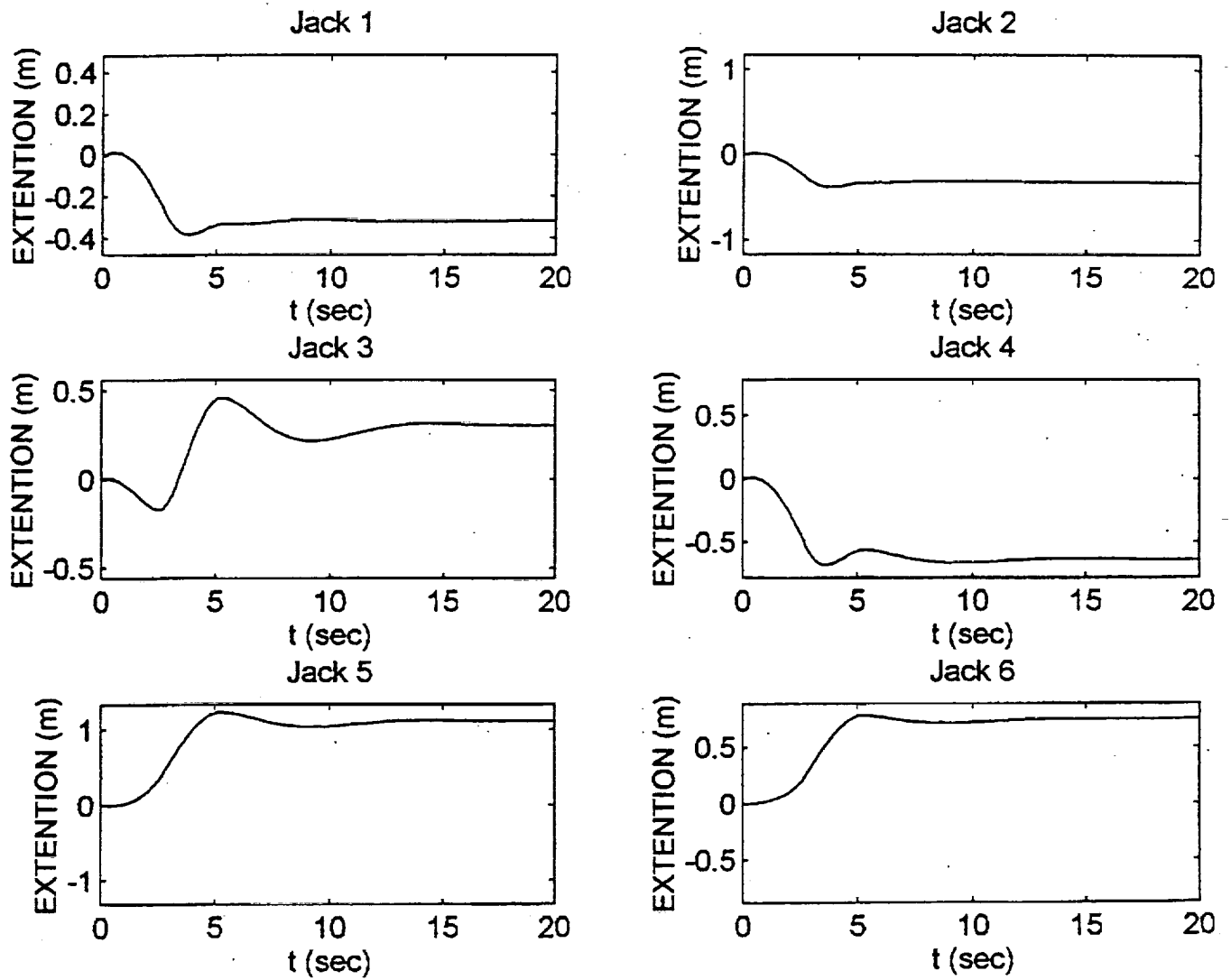


Figure B.2. Optimal Algorithm Roll Doublet Pulse Input.

NASA ROLL TEST CASE
DOUBLET INPUT

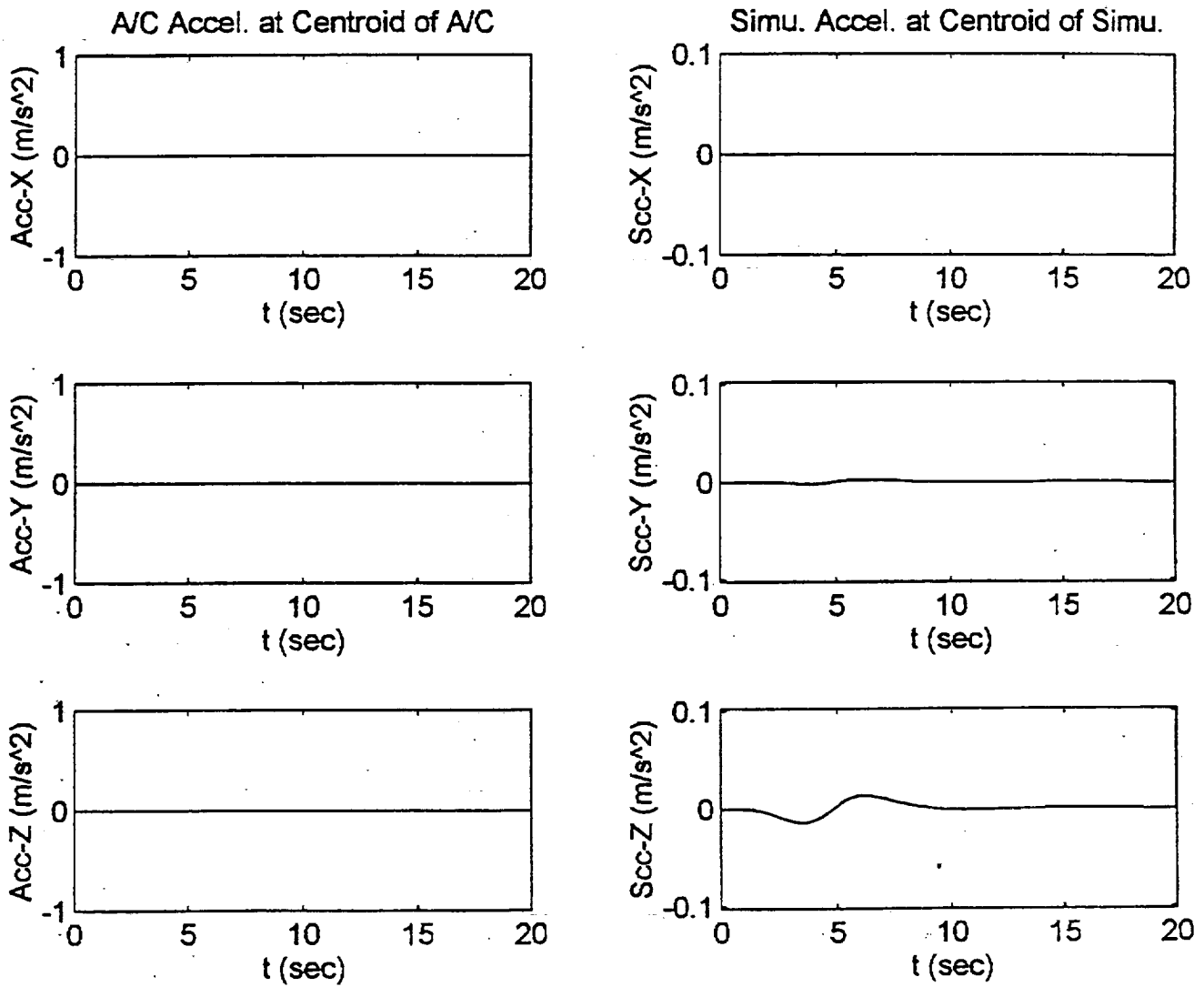


Figure B.3. NASA Adaptive Algorithm Roll Doublet Pulse Input.

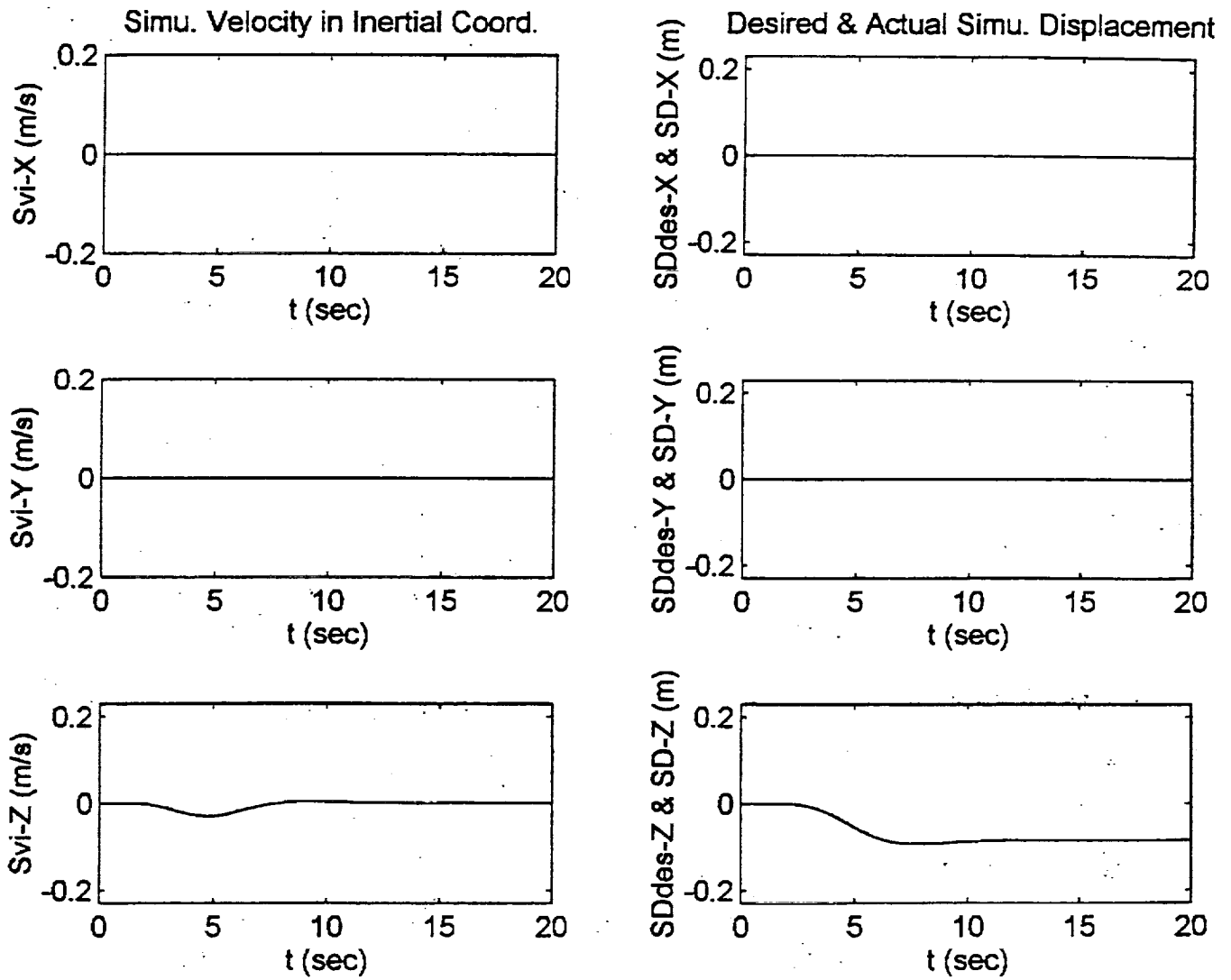


Figure B.3. NASA Adaptive Algorithm Roll Doublet Pulse Input.

NASA ROLL TEST CASE
DOUBLET INPUT

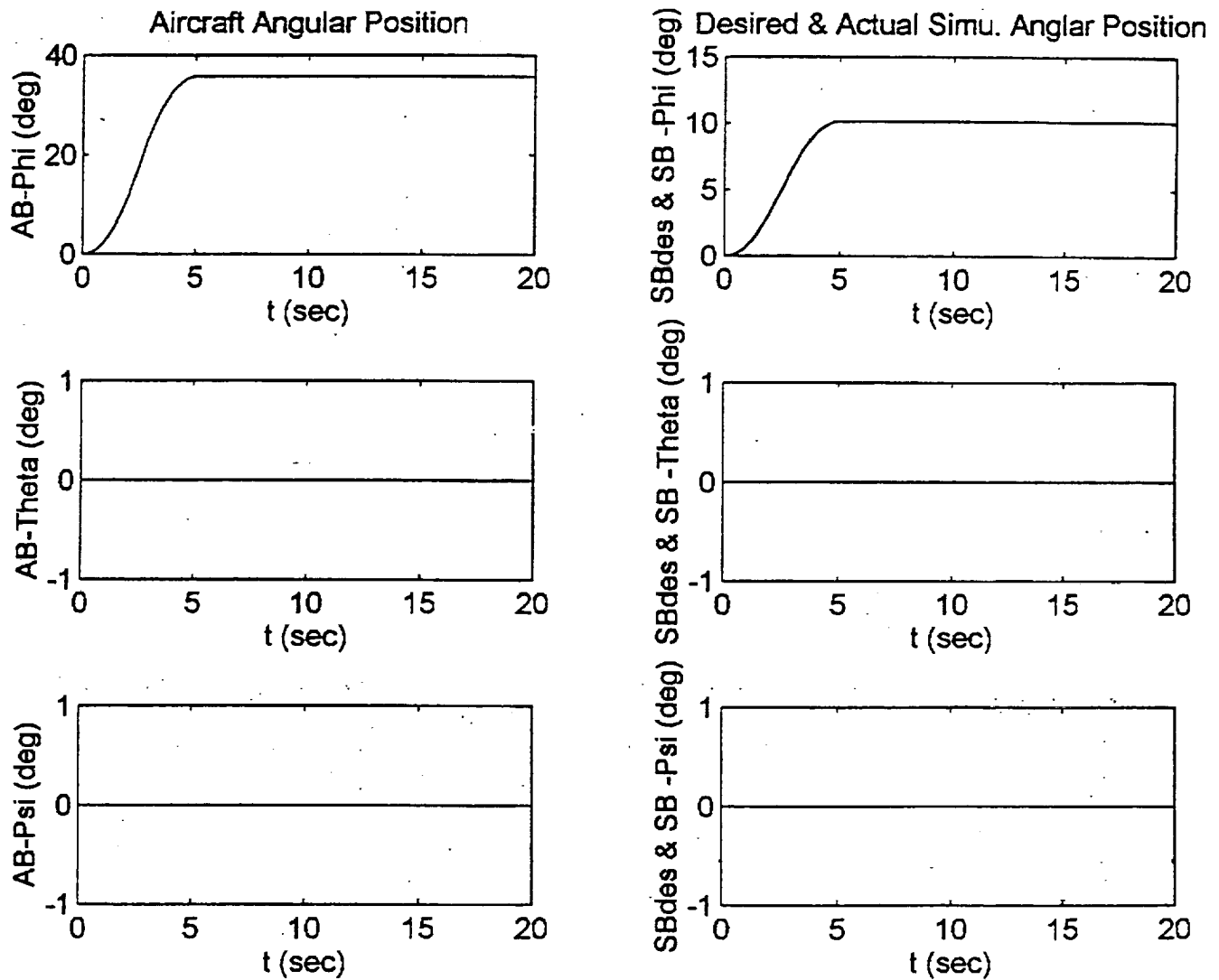


Figure B.3. NASA Adaptive Algorithm Roll Doublet Pulse Input.

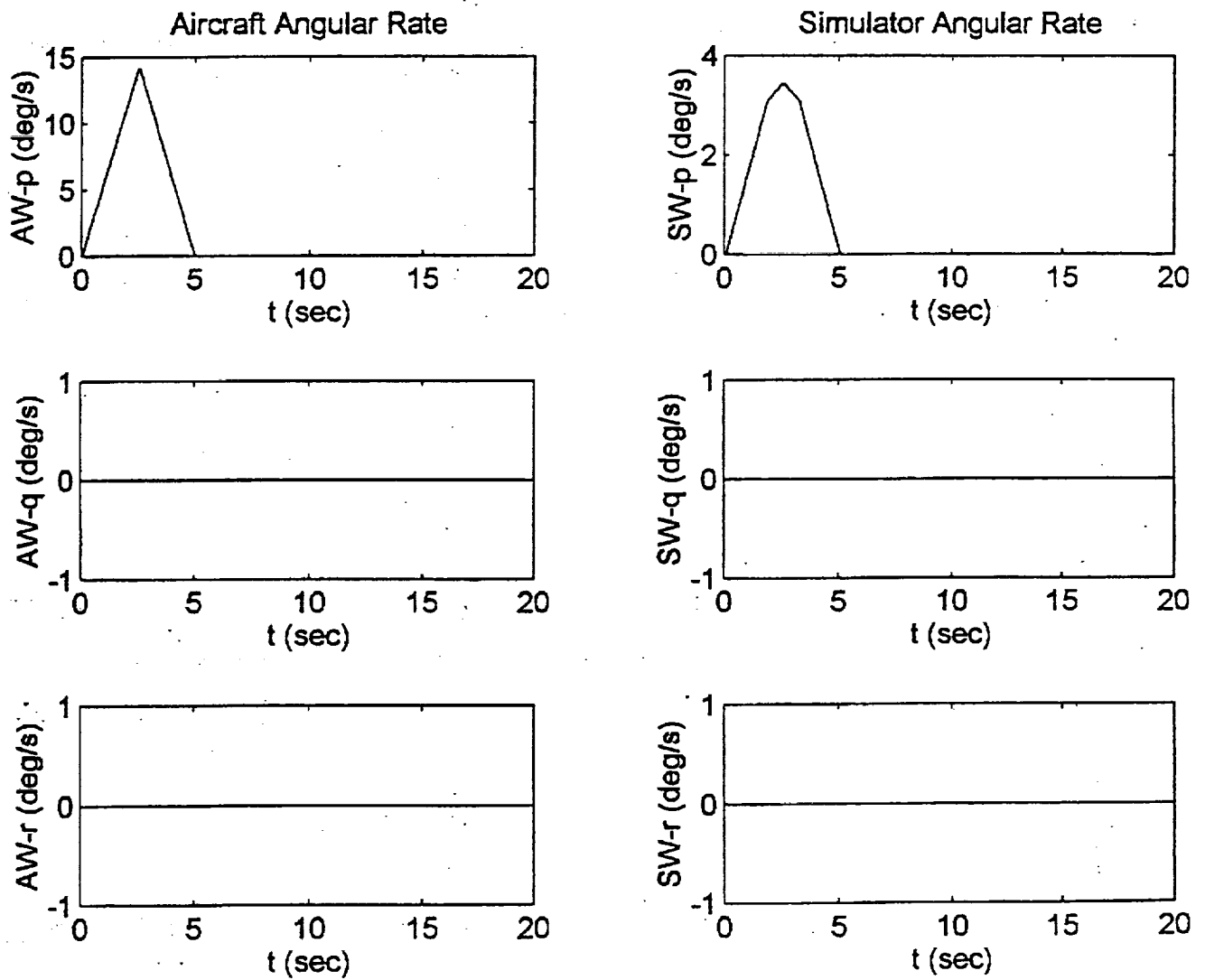


Figure B.3. NASA Adaptive Algorithm Roll Doublet Pulse Input.

NASA ROLL TEST CASE
DOUBLET INPUT

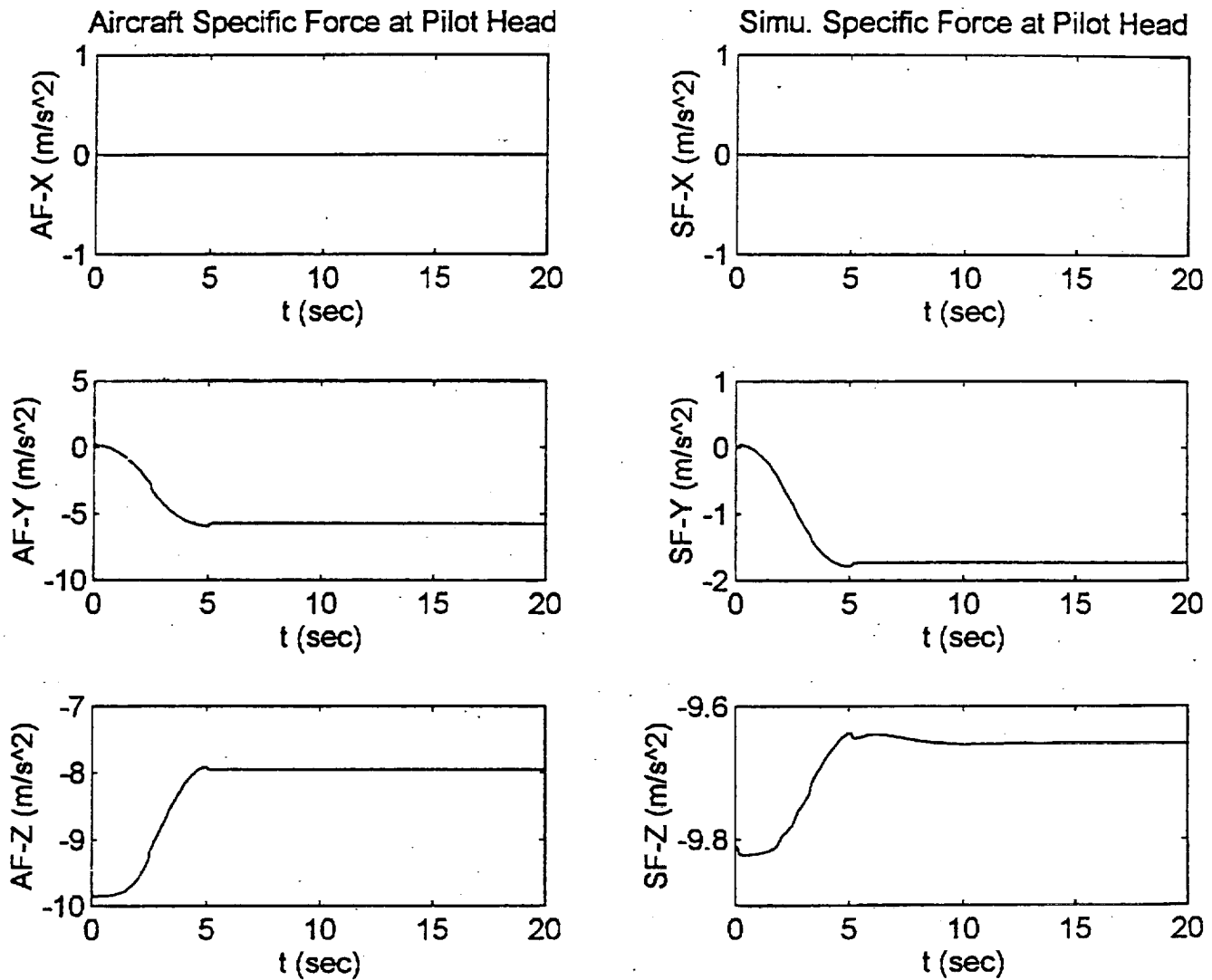


Figure B.3. NASA Adaptive Algorithm Roll Doublet Pulse Input.

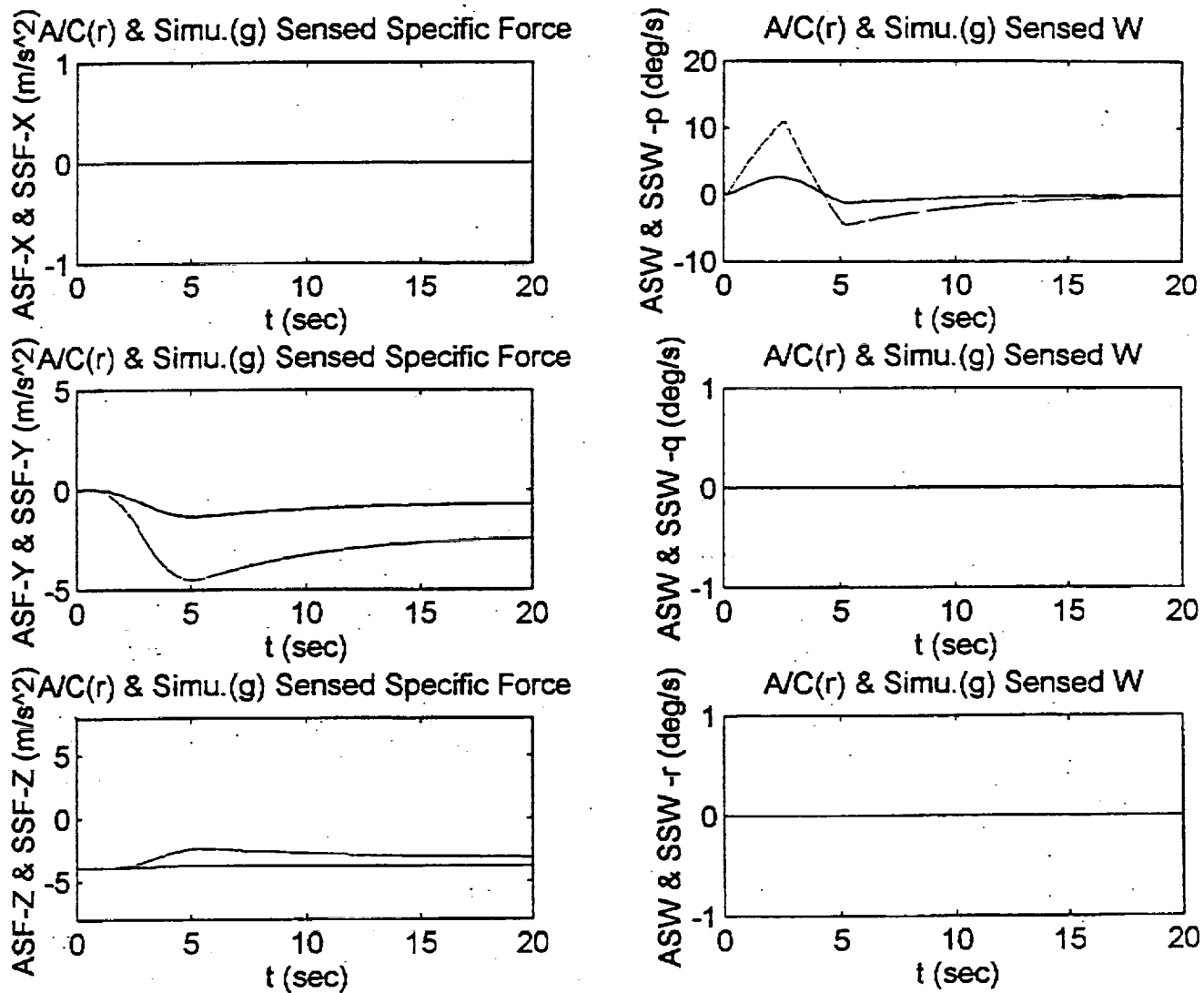


Figure B.3. NASA Adaptive Algorithm Roll Doublet Pulse Input.

NASA ROLL TEST CASE
DOUBLET INPUT

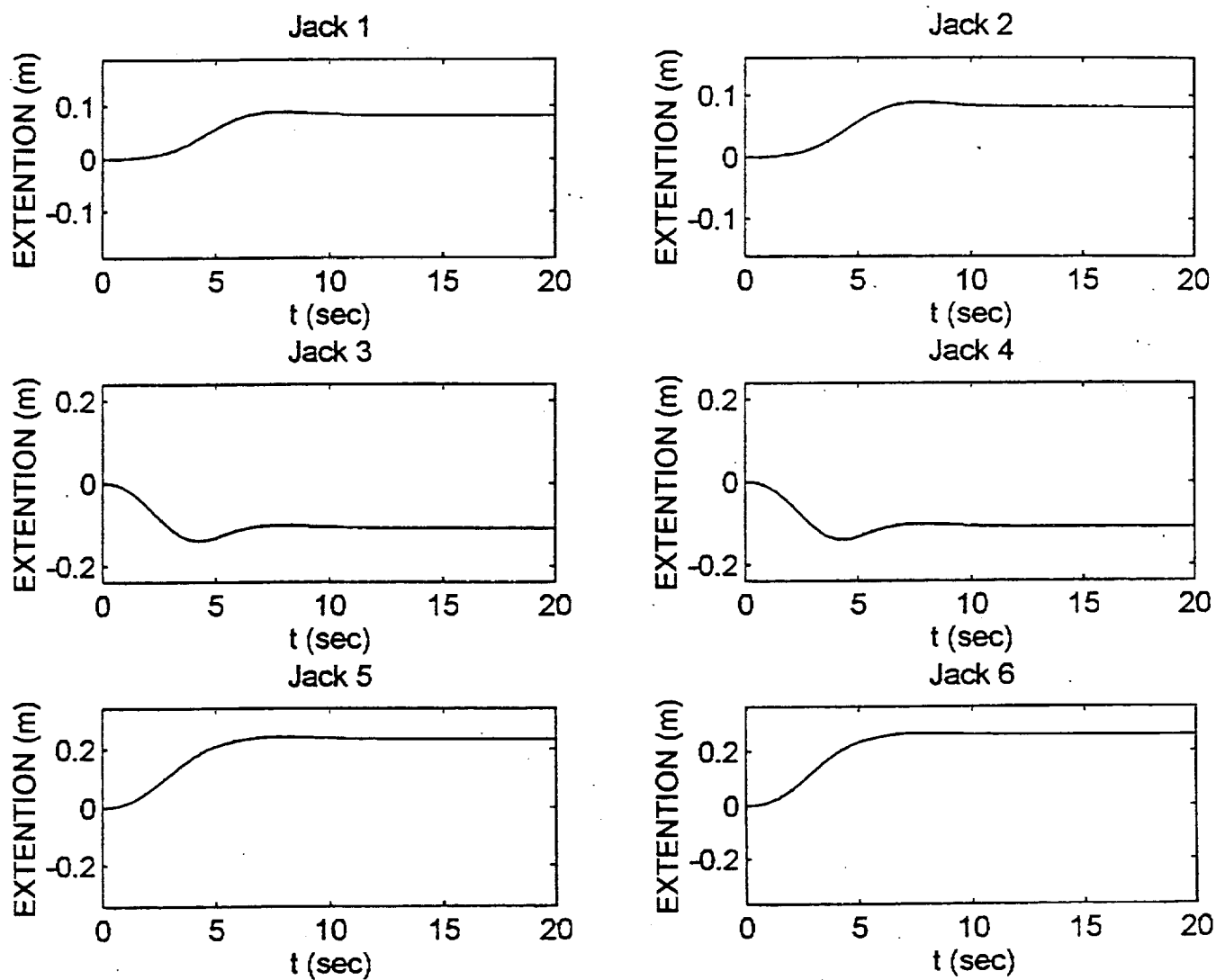


Figure B.3. NASA Adaptive Algorithm Roll Doublet Pulse Input.

ADAPTIVE ROLL TEST CASE
DOUBLET INPUT

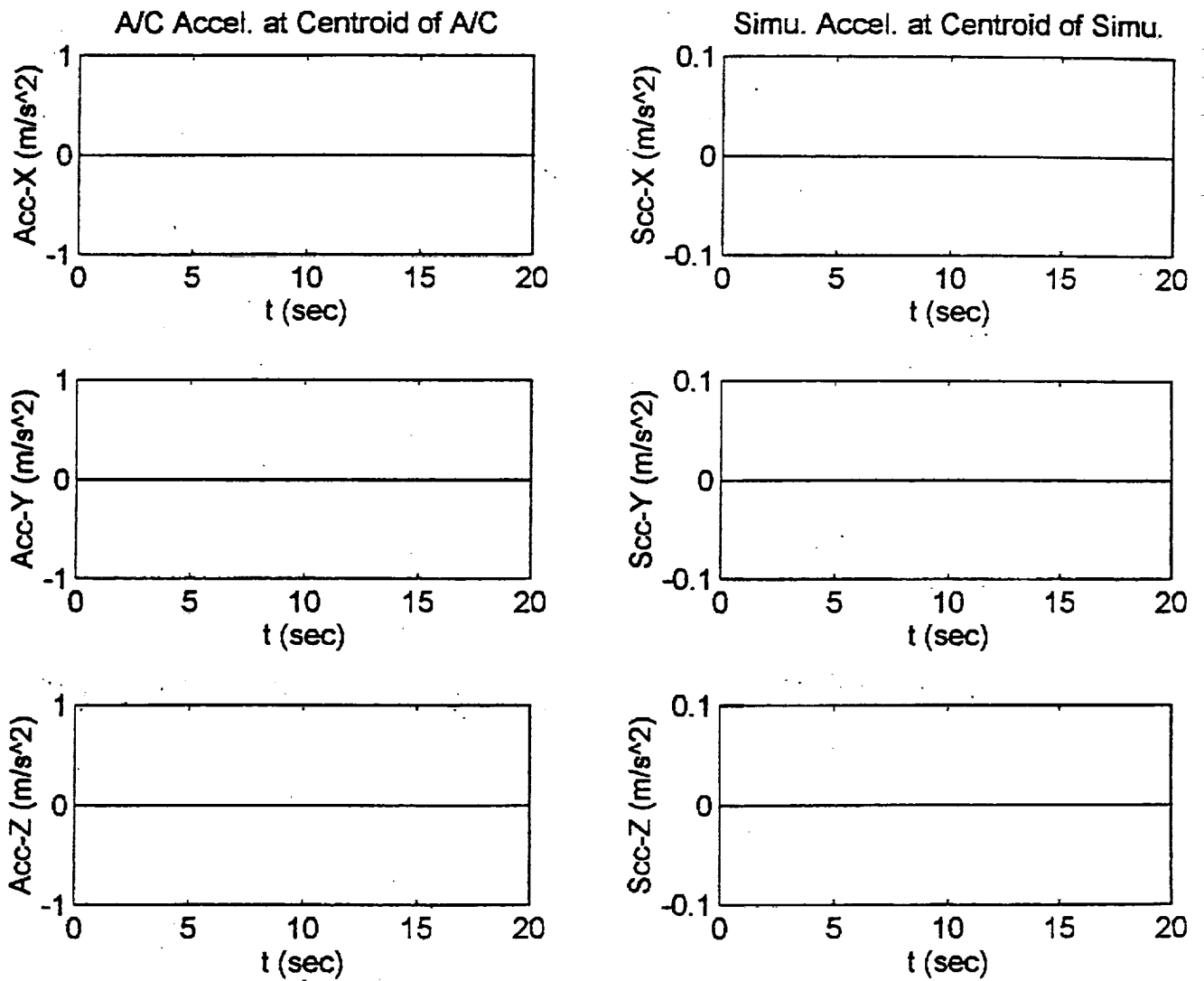


Figure B.4. UTIAS Adaptive Algorithm Roll Doublet Pulse Input.

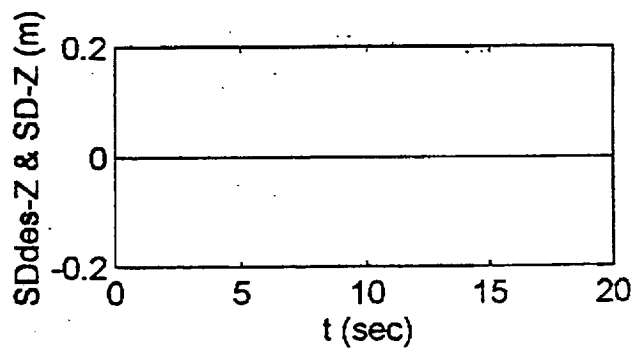
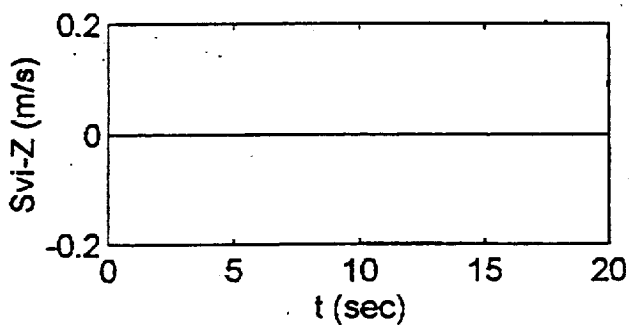
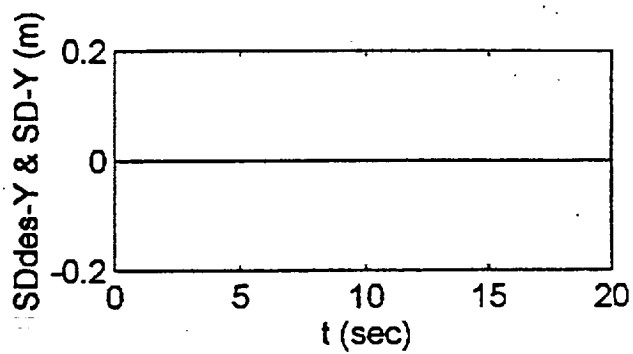
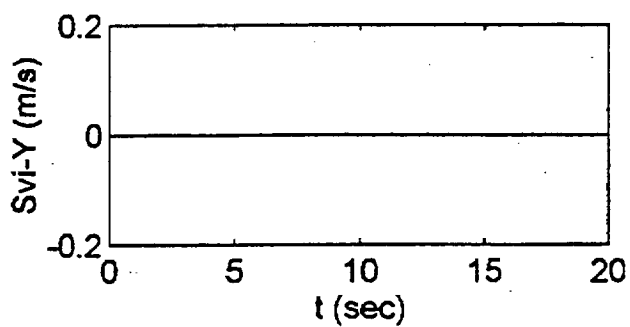
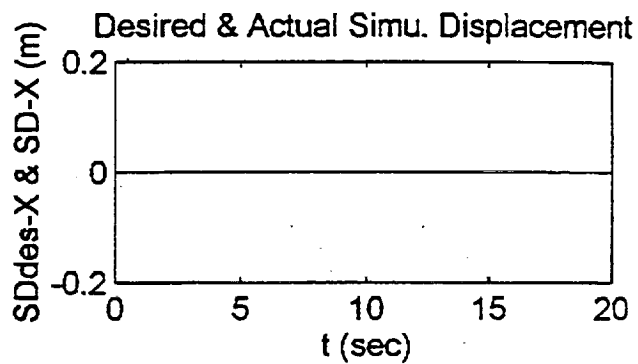
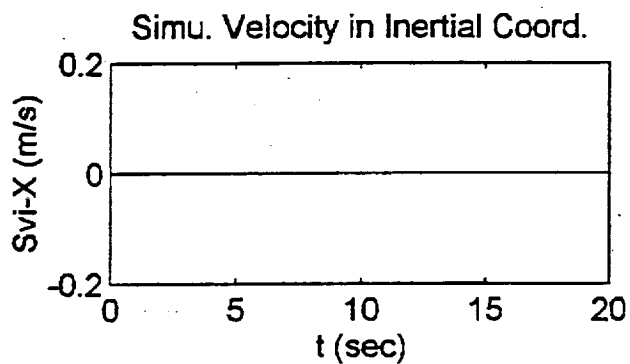


Figure B.4. UTIAS Adaptive Algorithm Roll Doublet Pulse Input.

ADAPTIVE ROLL TEST CASE
DOUBLET INPUT

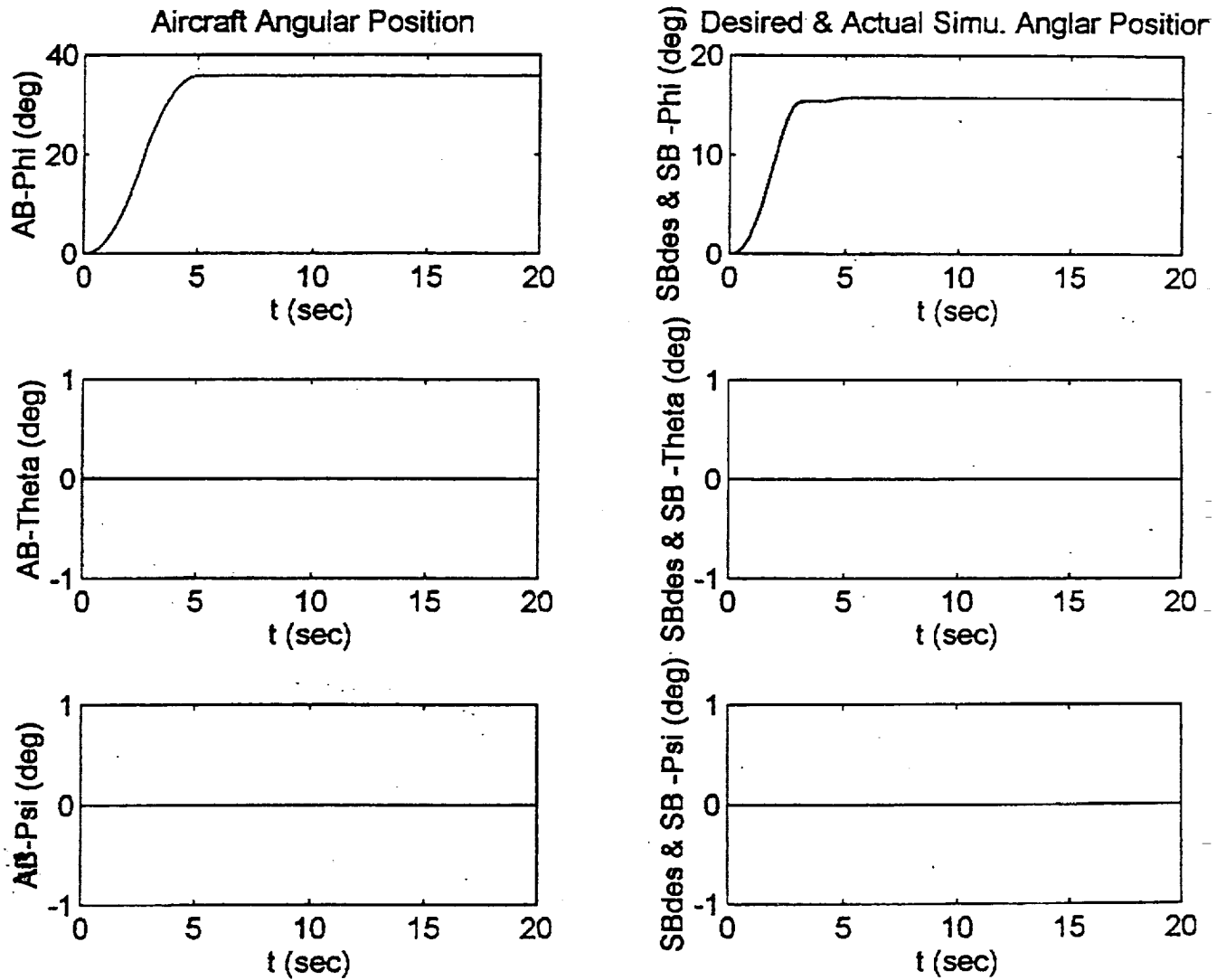


Figure B.4. UTIAS Adaptive Algorithm Roll Doublet Pulse Input.

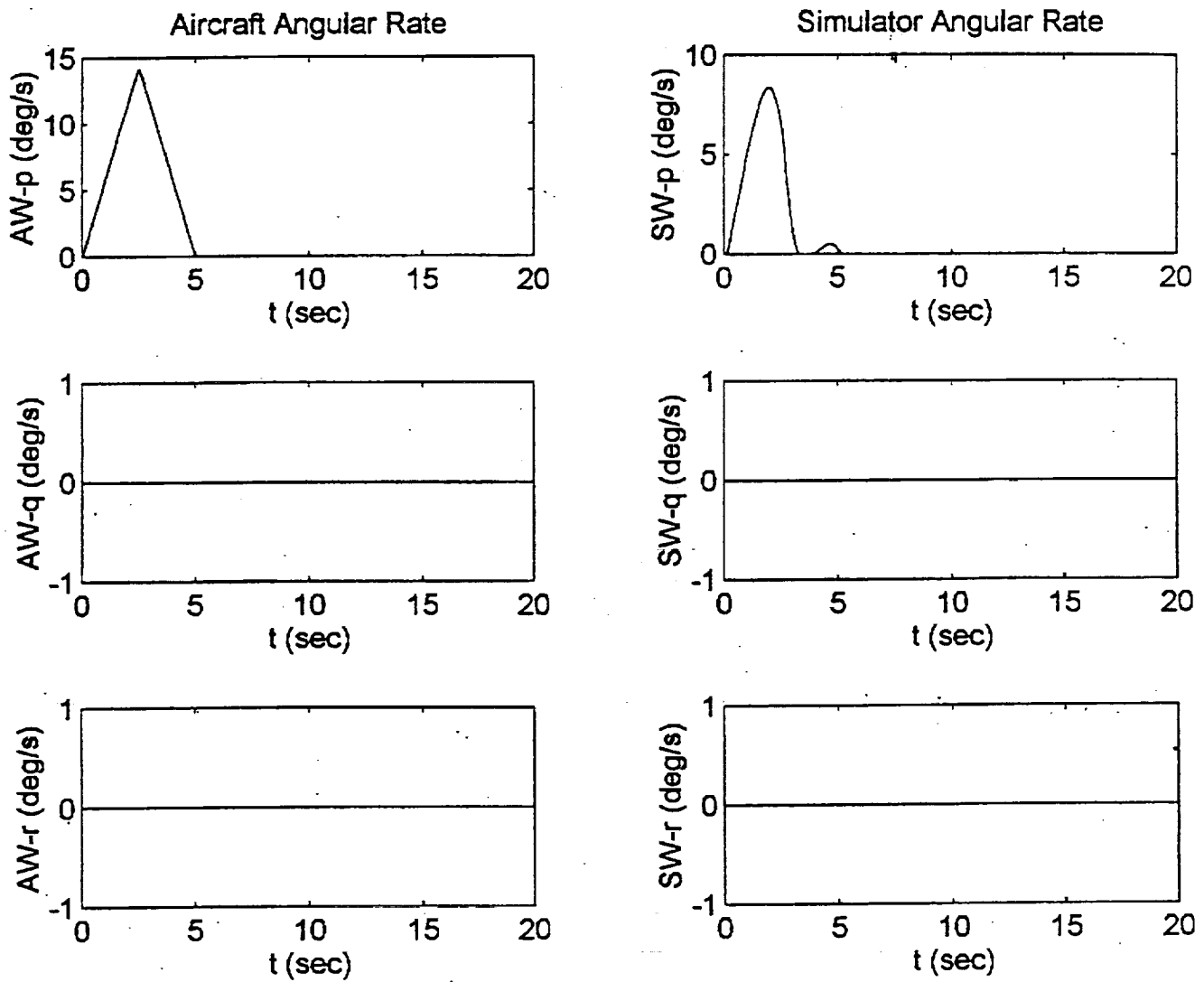


Figure B.4. UTIAS Adaptive Algorithm Roll Doublet Pulse Input.

ADAPTIVE ROLL TEST CASE
DOUBLET INPUT

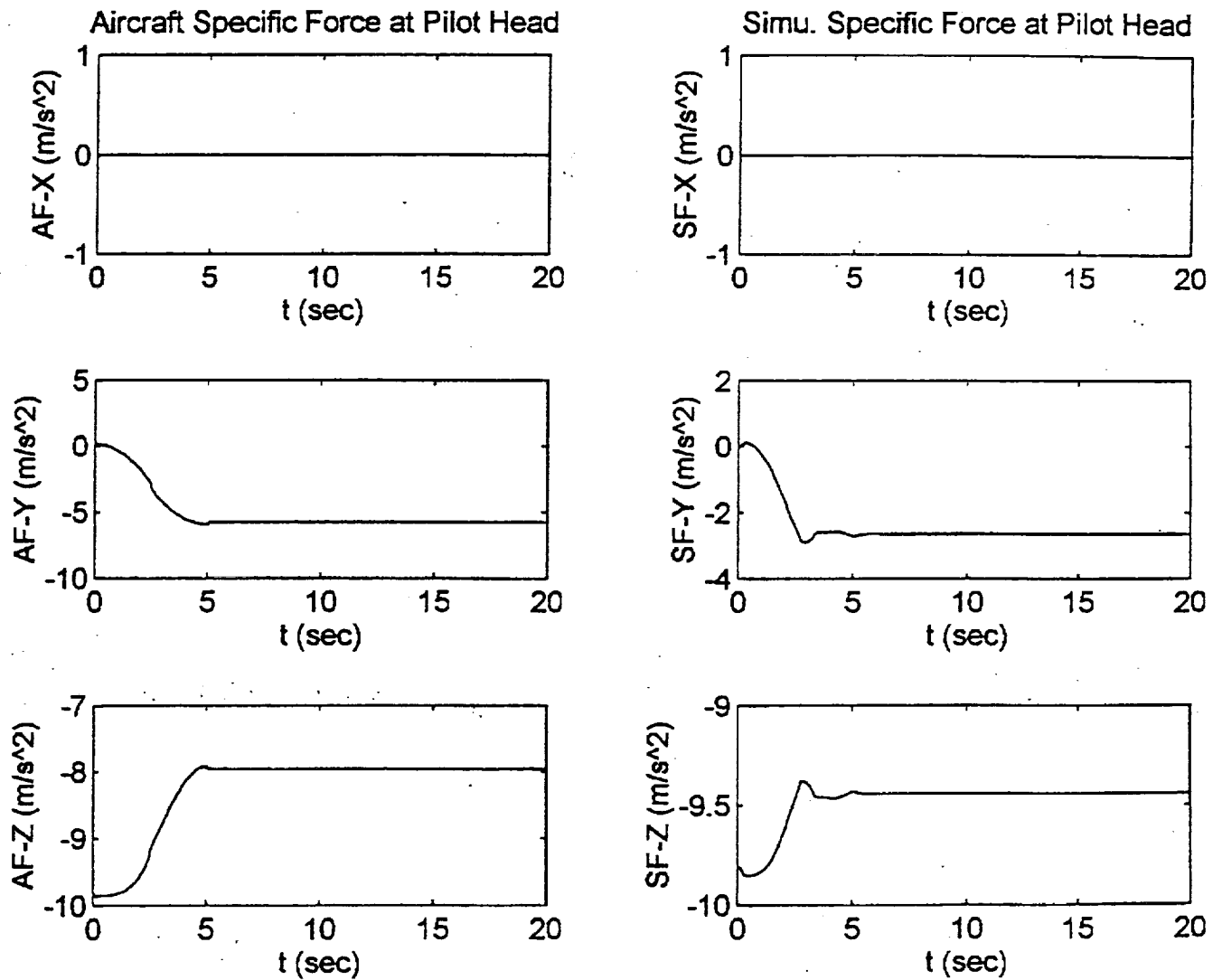


Figure B.4. UTIAS Adaptive Algorithm Roll Doublet Pulse Input.

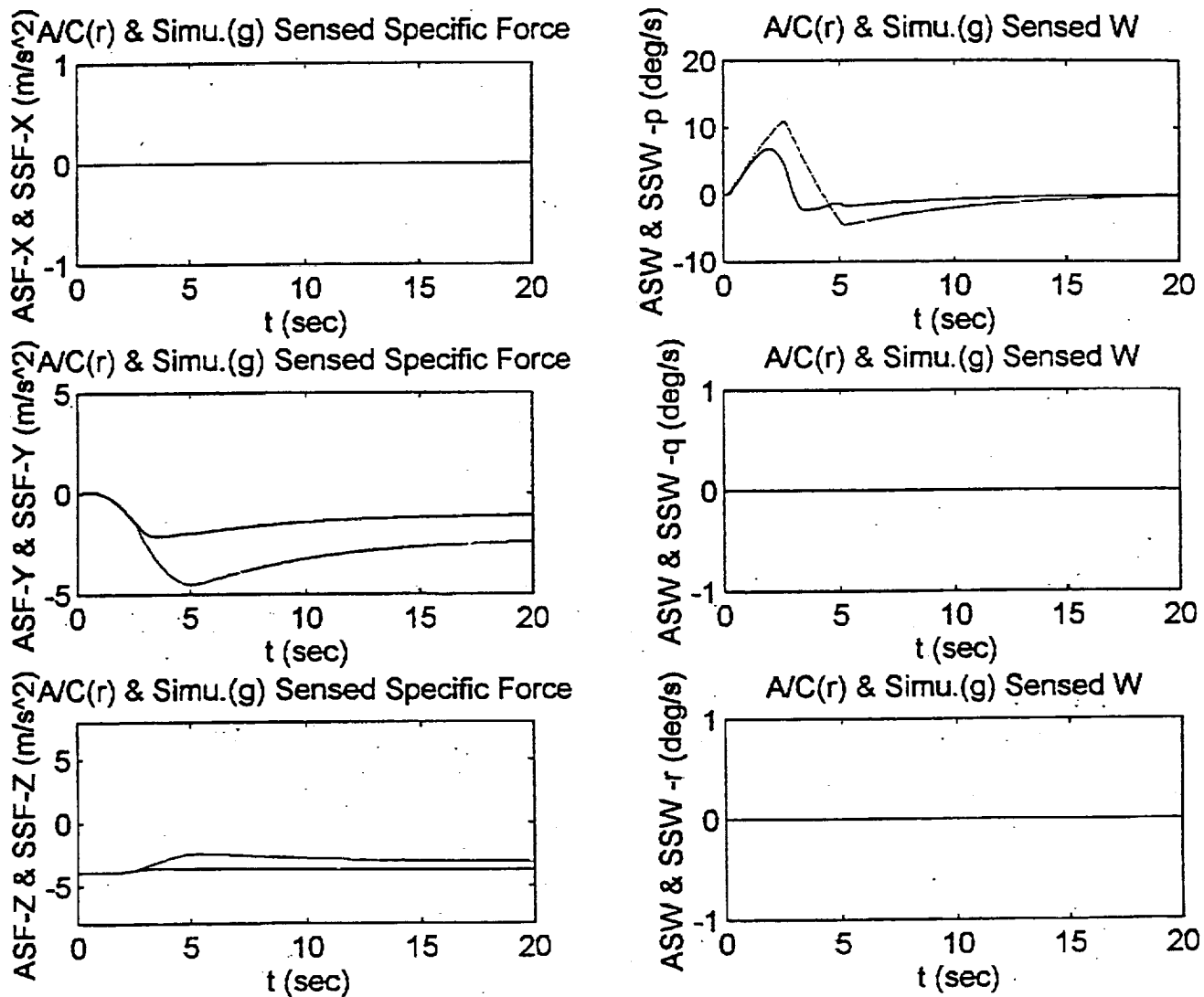


Figure B.4. UTIAS Adaptive Algorithm Roll Doublet Pulse Input.

ADAPTIVE ROLL TEST CASE
DOUBLET INPUT

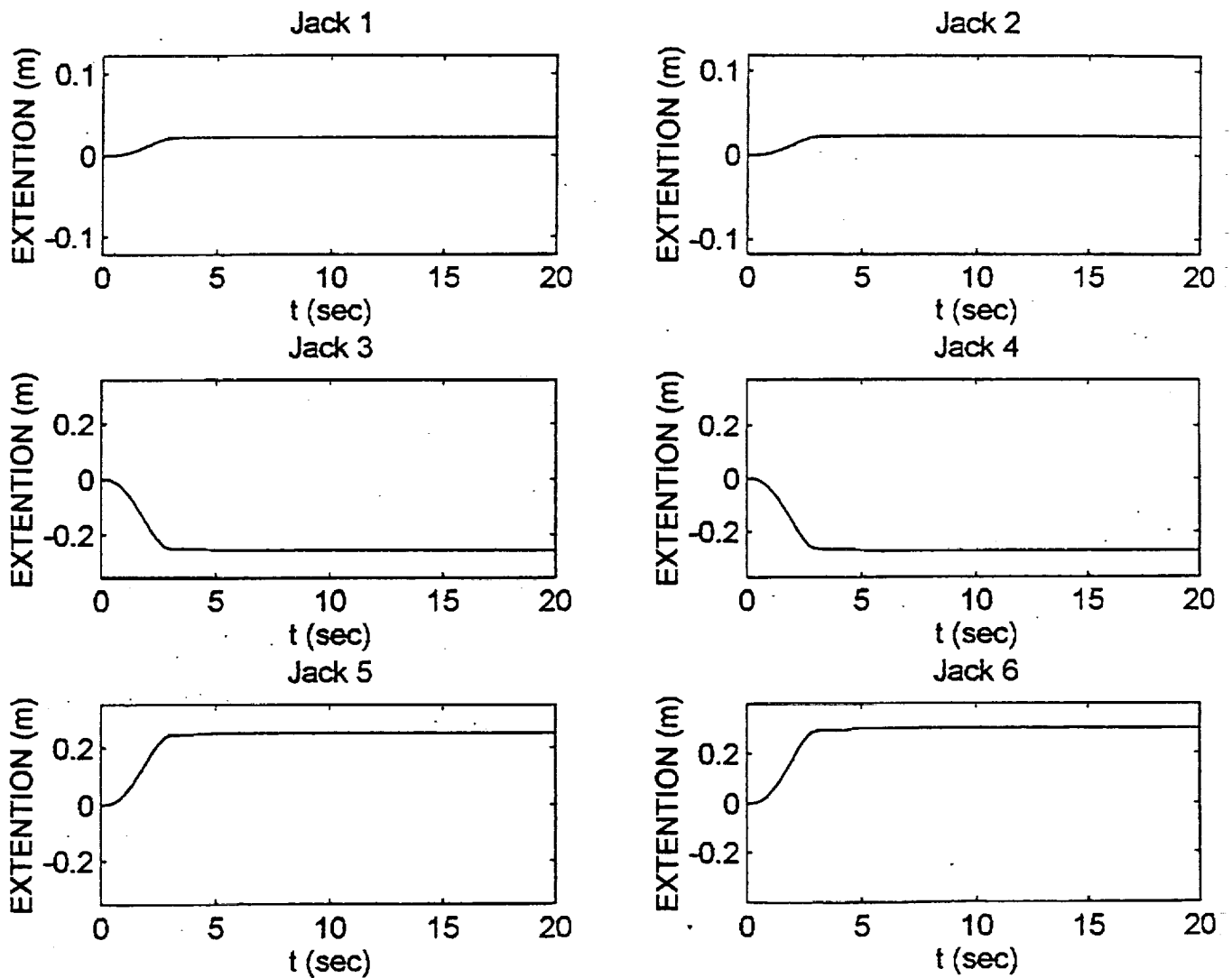


Figure B.4. UTIAS Adaptive Algorithm Roll Doublet Pulse Input.

Appendix C.

Phase 2 Test Run Output Figures

- Figure C.1. Optimal Algorithm Surge Half Sinusoidal Pulse Input.
0.1 Hz half cycle, 1 m/s^2 peak x-acceleration input.
- Figure C.2. NASA Adaptive Algorithm Surge Half Sinusoidal Pulse Input.
0.1 Hz half cycle, 1 m/s^2 peak x-acceleration input.
- Figure C.3. Surge Pulse Input Specific Force Comparison.
Comparison of the specific forces for both algorithms.
- Figure C.4. Optimal Algorithm Surge $3 \text{ m/s}^2/\text{s}$ Ramp-Step Input.
 $3 \text{ m/s}^2/\text{s}$ slope, 5 m/s^2 peak x-acceleration input.
- Figure C.5. NASA Adaptive Algorithm Surge $3 \text{ m/s}^2/\text{s}$ Ramp-Step Input.
 $3 \text{ m/s}^2/\text{s}$ slope, 5 m/s^2 peak x-acceleration input.
- Figure C.6. Surge $3 \text{ m/s}^2/\text{s}$ Ramp-Step Input Specific Force Comparison.
Comparison of the specific forces for both algorithms.
- Figure C.7. Optimal Algorithm Surge $5 \text{ m/s}^2/\text{s}$ Ramp-Step Input.
 $5 \text{ m/s}^2/\text{s}$ slope, 5 m/s^2 peak x-acceleration input.
- Figure C.8. NASA Adaptive Algorithm Surge $5 \text{ m/s}^2/\text{s}$ Ramp-Step Input.
 $5 \text{ m/s}^2/\text{s}$ slope, 5 m/s^2 peak x-acceleration input.
- Figure C.9. Surge $5 \text{ m/s}^2/\text{s}$ Ramp-Step Input Specific Force Comparison.
Comparison of the specific forces for both algorithms.
- Figure C.10. Optimal Algorithm Sway Half Sinusoidal Pulse Input.
0.5 Hz half cycle, 1 m/s^2 peak y-acceleration input.
- Figure C.11. NASA Adaptive Algorithm Sway Half Sinusoidal Pulse Input.
0.5 Hz half cycle, 1 m/s^2 peak y-acceleration input.
- Figure C.12. Sway Pulse Input Specific Force Comparison.
Comparison of the specific forces for both algorithms.

- Figure C.13. Optimal Algorithm Heave Square Pulse Input.
3 m/s² peak, 5 second duration z-acceleration input.
- Figure C.14. NASA Adaptive Algorithm Heave Square Pulse Input.
3 m/s² peak, 5 second duration z-acceleration input.
- Figure C.15. Heave Square Pulse Input Specific Force Comparison.
Comparison of the specific forces for both algorithms.
- Figure C.16. Optimal Algorithm Roll Doublet Pulse Input.
0.1 rad/sec² peak, 5 second duration input.
- Figure C.17. NASA Adaptive Algorithm Roll Doublet Pulse Input.
0.1 rad/sec² peak, 5 second duration input.
- Figure C.18. Roll Doublet Specific Force Comparison.
Comparison of the specific forces for both algorithms.
- Figure C.19. Optimal Algorithm Pitch Doublet Pulse Input.
0.1 rad/sec² peak, 5 second duration input.
- Figure C.20. NASA Adaptive Algorithm Pitch Doublet Pulse Input.
0.1 rad/sec² peak, 5 second duration input.
- Figure C.21. Pitch Doublet Specific Force Comparison.
Comparison of the specific forces for both algorithms.
- Figure C.22. Optimal Algorithm Yaw Doublet Pulse Input.
0.1 rad/sec² peak, 5 second duration input.
- Figure C.23. NASA Adaptive Algorithm Yaw Doublet Pulse Input.
0.1 rad/sec² peak, 5 second duration input.
- Figure C.24. Yaw Doublet Specific Force Comparison.
Comparison of the specific forces for both algorithms.
- Figure C.25. Sway Pulse Input Braking Algorithm Test.
10 m/s² peak, 10 second duration y-acceleration input.

Specific force comparison figures are labelled as follows:

Line with no marks – aircraft response.

Line with marks – simulator response: o – NASA Adaptive; * – Optimal

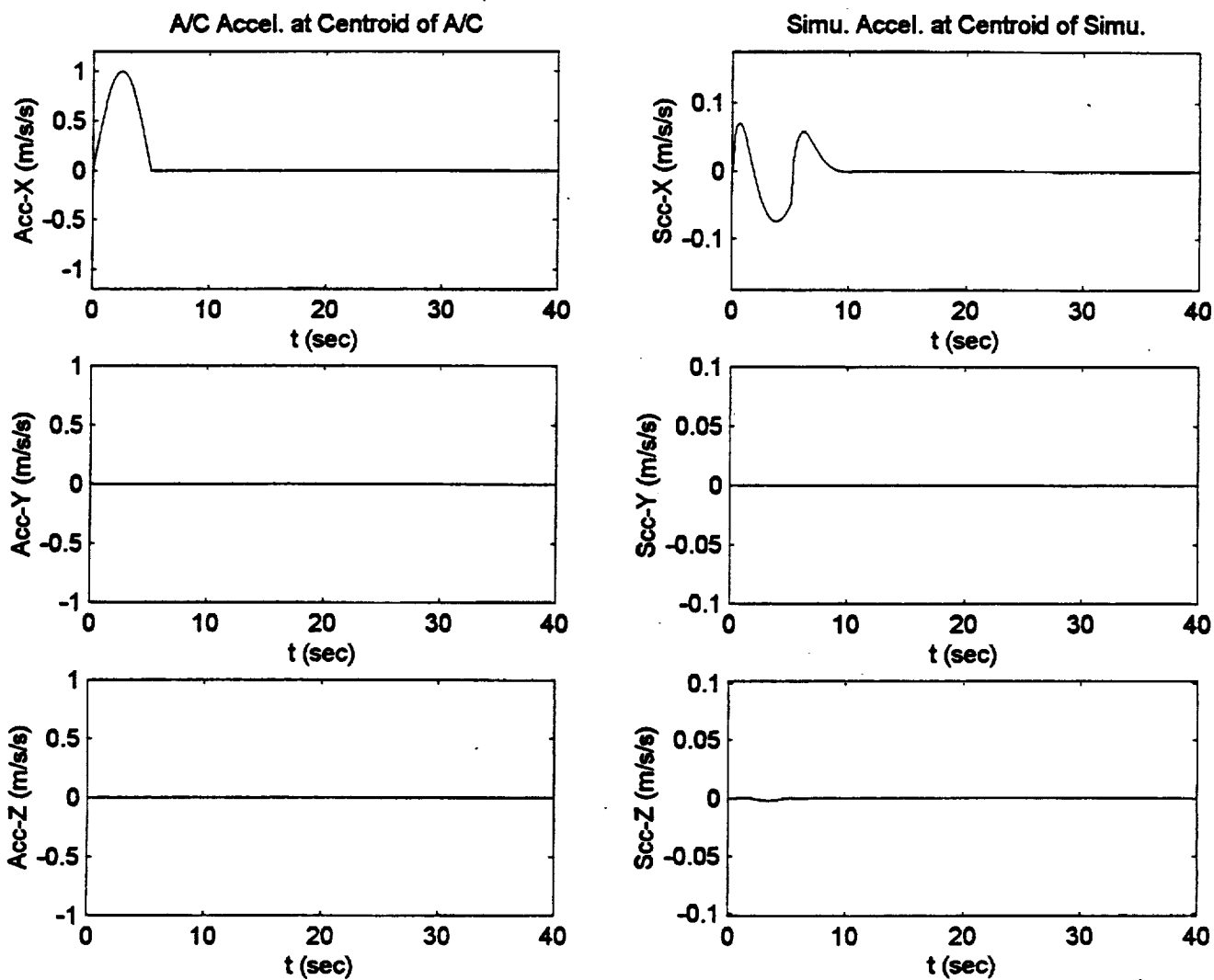


Figure C.1. Optimal Algorithm Surge Half Sinusoidal Pulse Input.

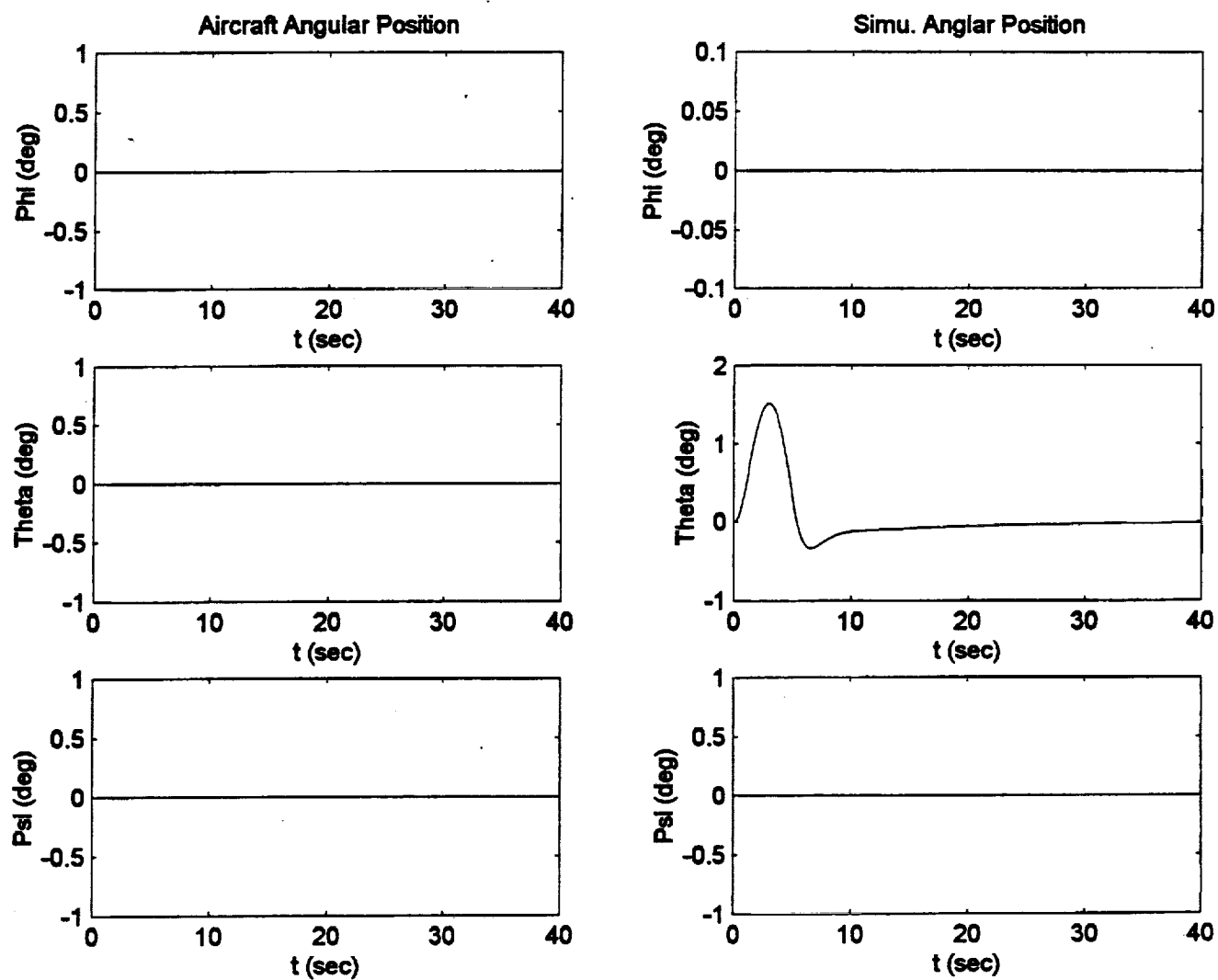


Figure C.1. Optimal Algorithm Surge Half Sinusoidal Pulse Input.

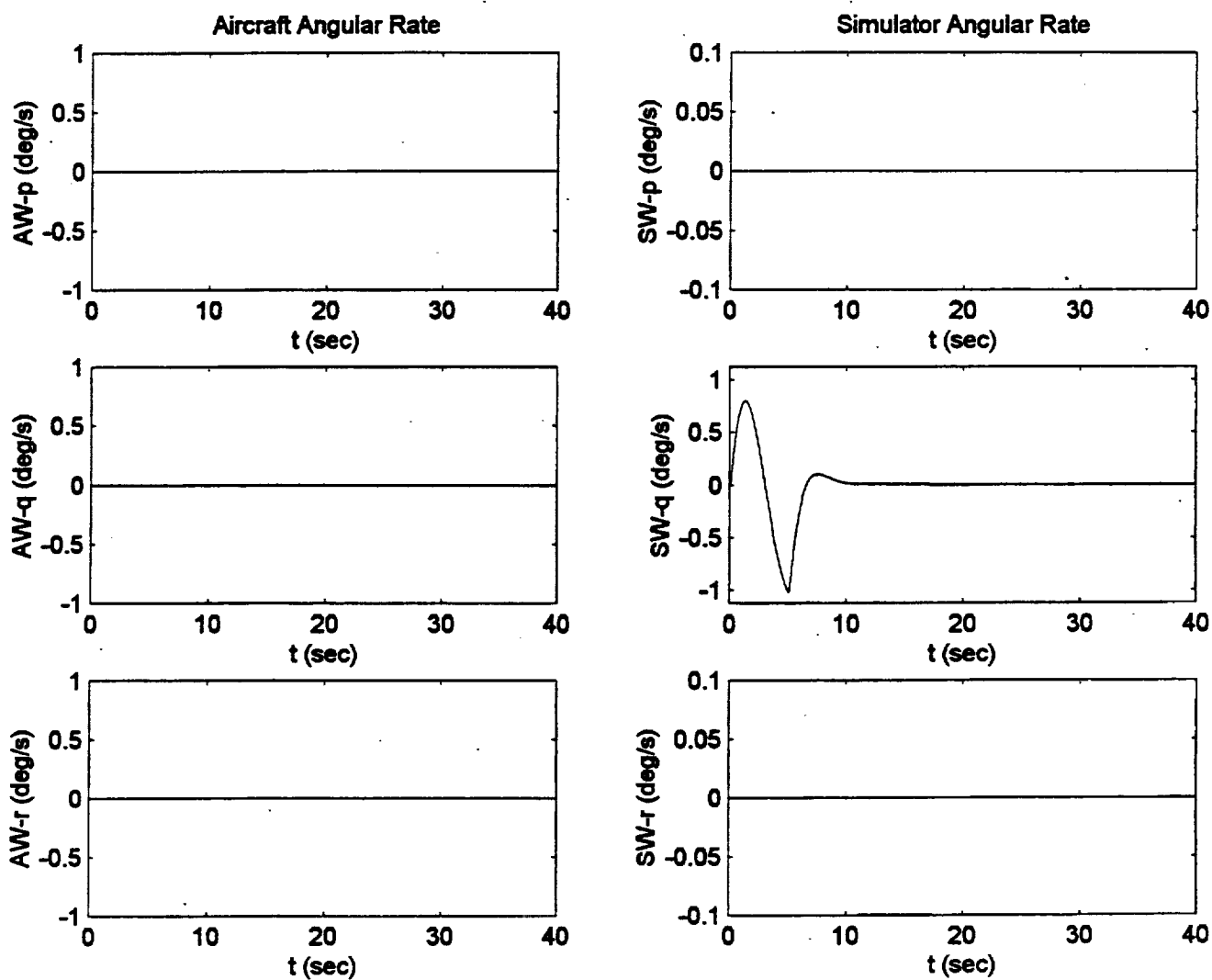


Figure C.1. Optimal Algorithm Surge Half Sinusoidal Pulse Input.

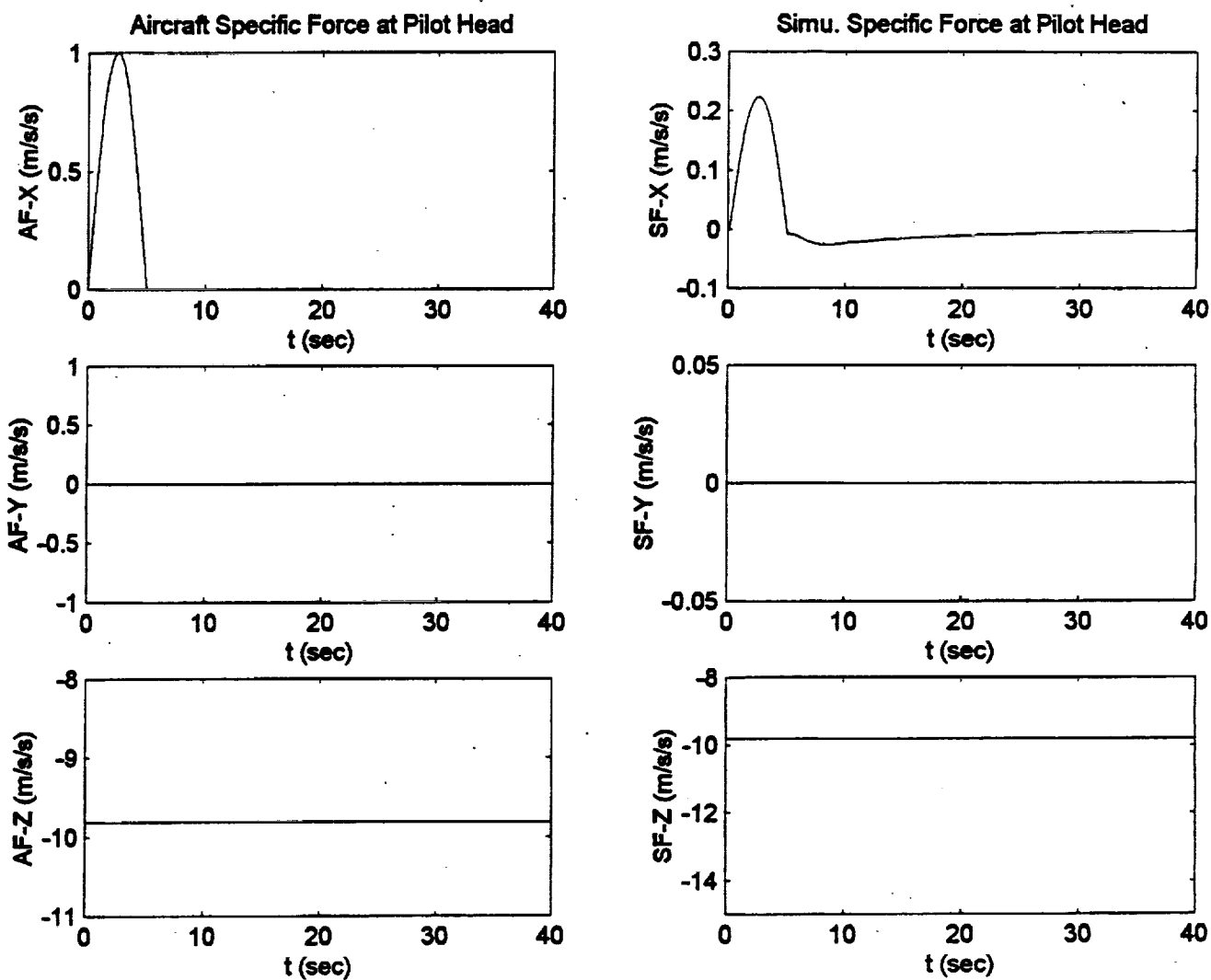


Figure C.1. Optimal Algorithm Surge Half Sinusoidal Pulse Input.

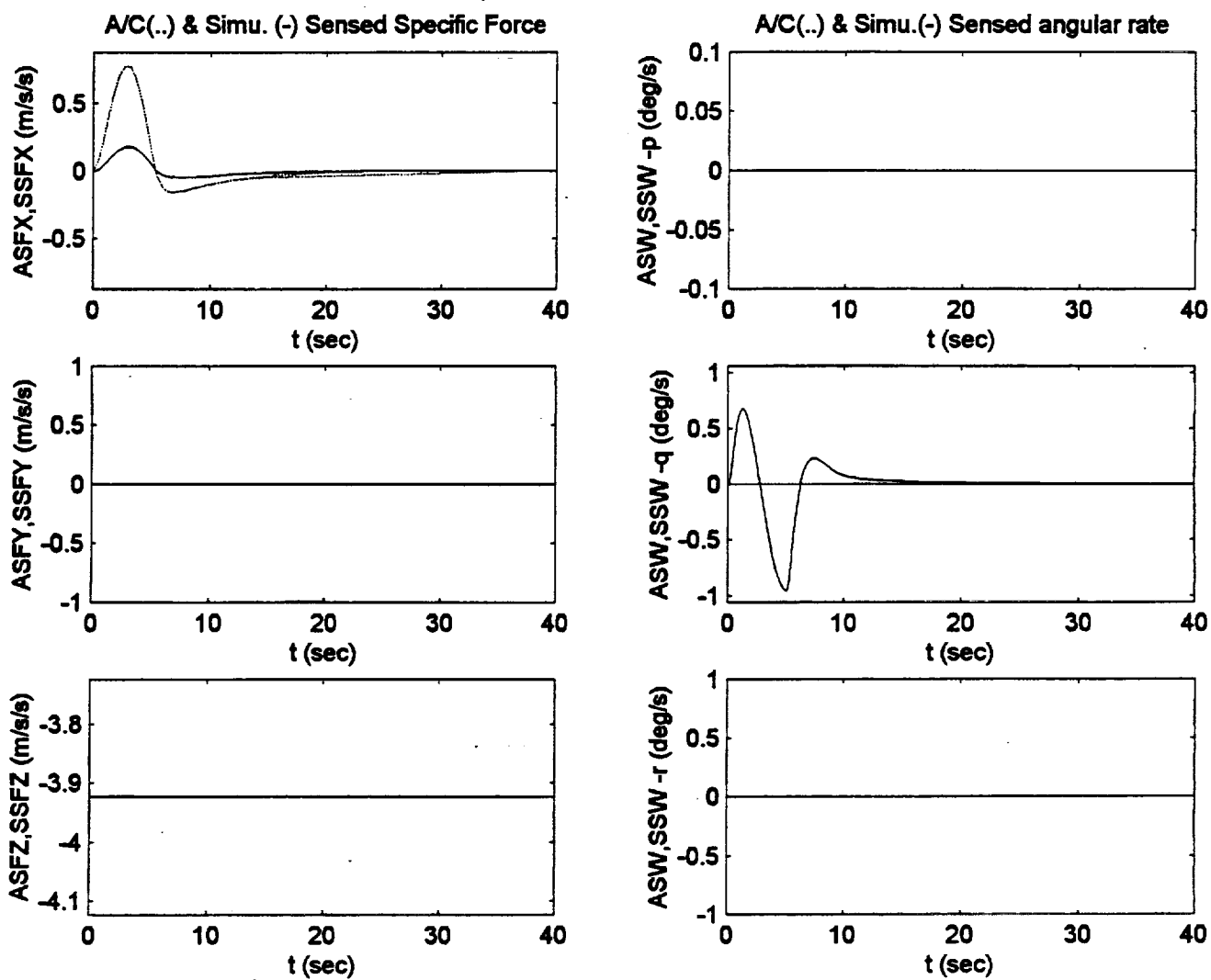


Figure C.1. Optimal Algorithm Surge Half Sinusoidal Pulse Input.

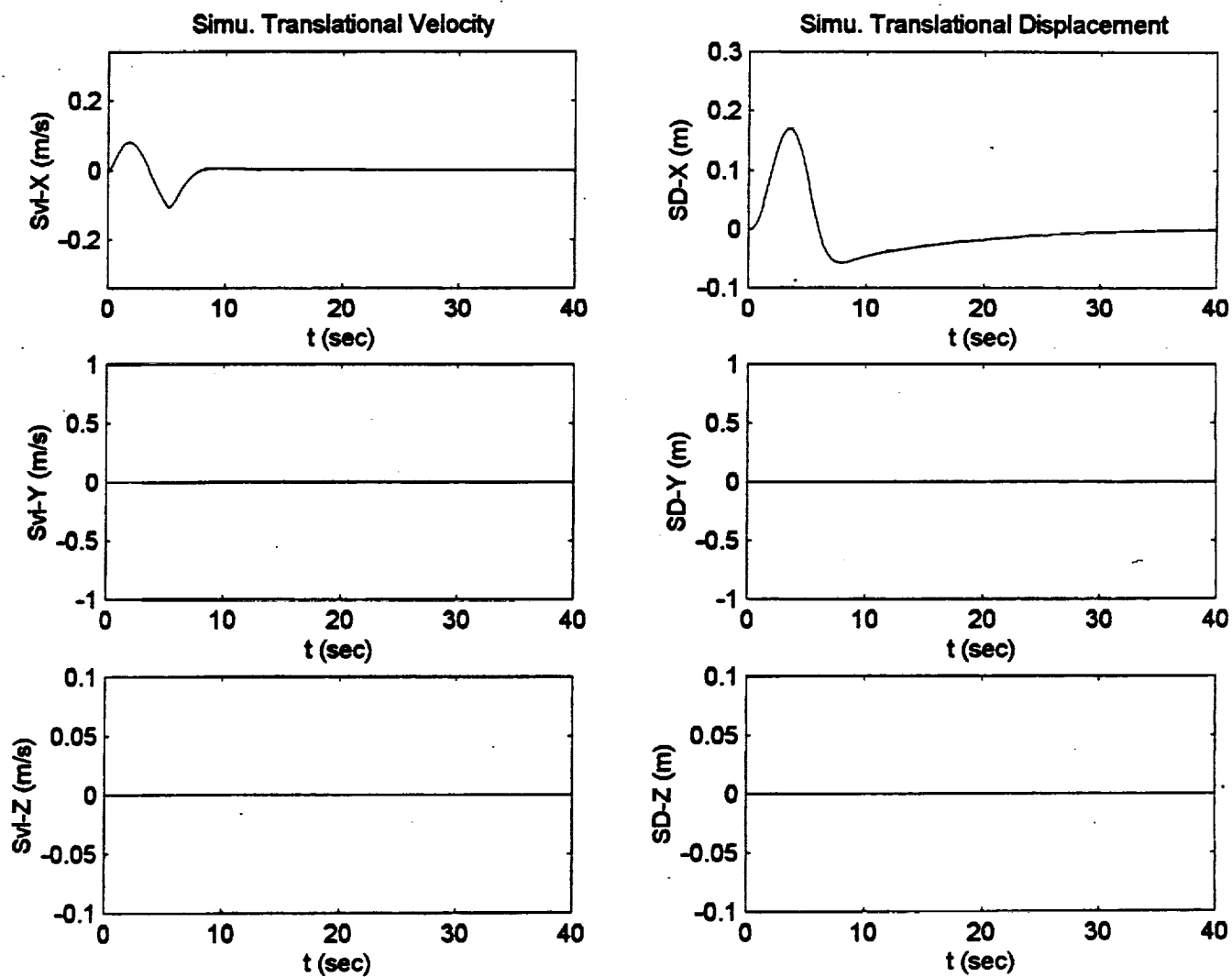


Figure C.1. Optimal Algorithm Surge Half Sinusoidal Pulse Input.

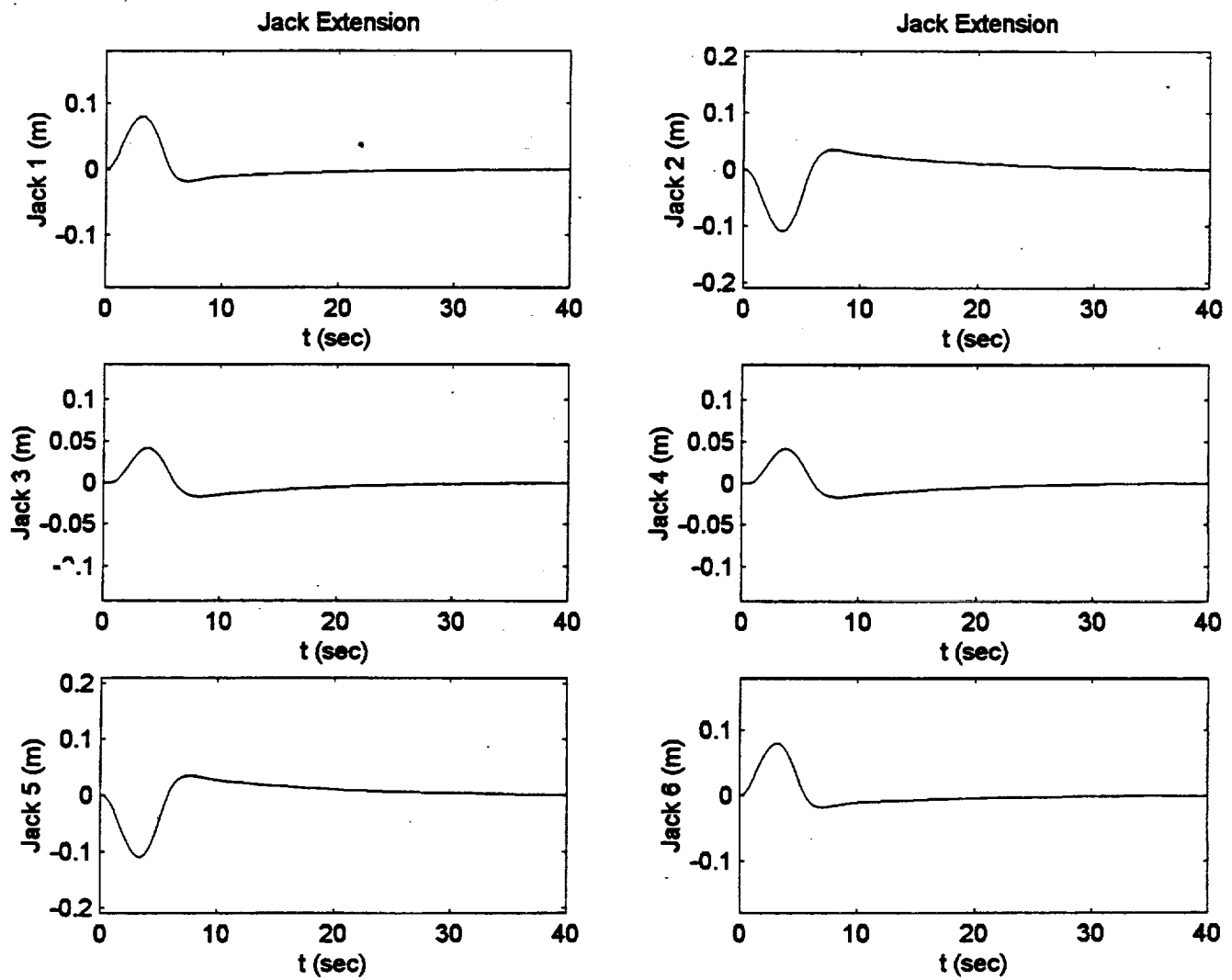


Figure C.1. Optimal Algorithm Surge Half Sinusoidal Pulse Input.

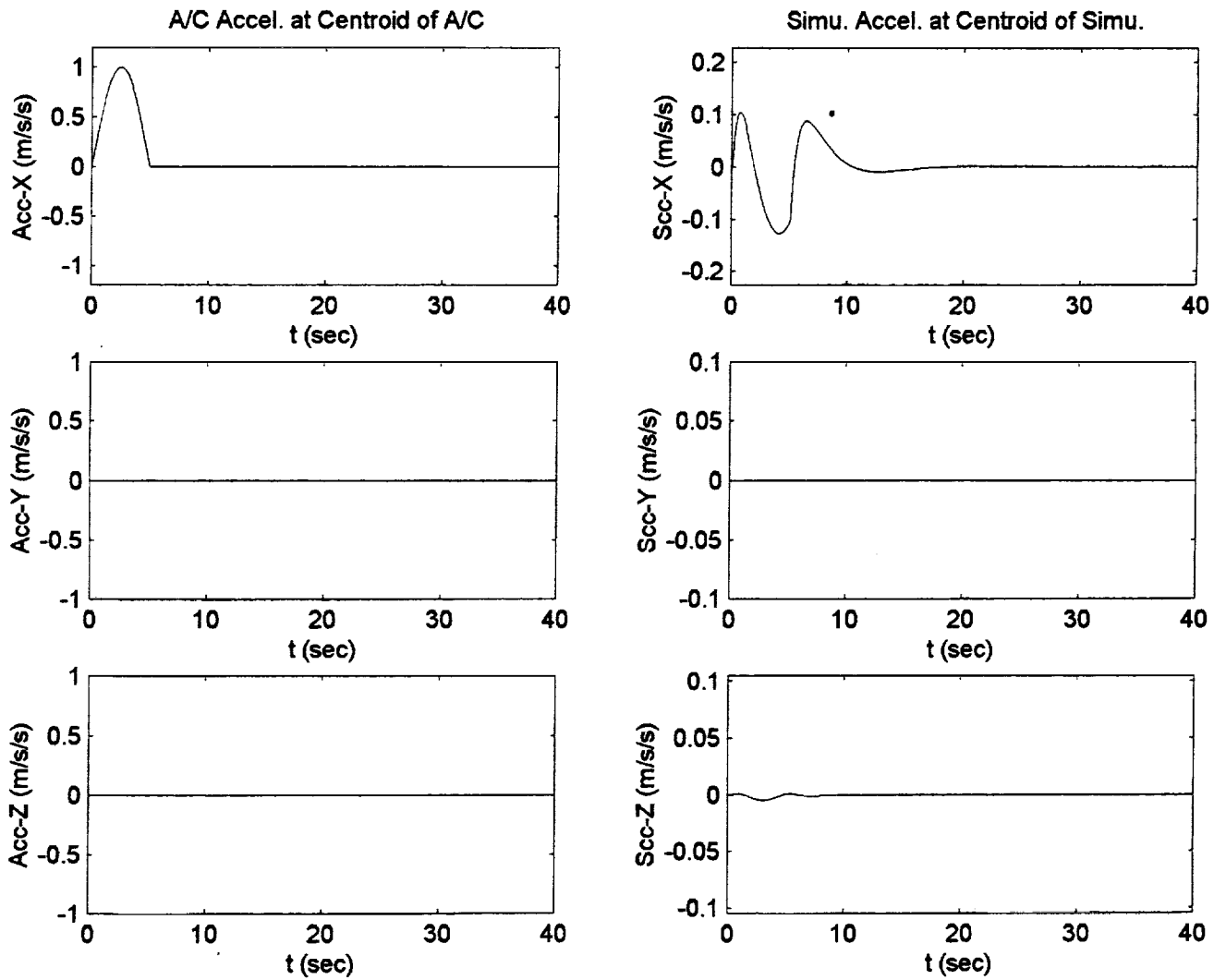


Figure C.2. NASA Adaptive Algorithm Surge Half Sinusoidal Pulse Input.

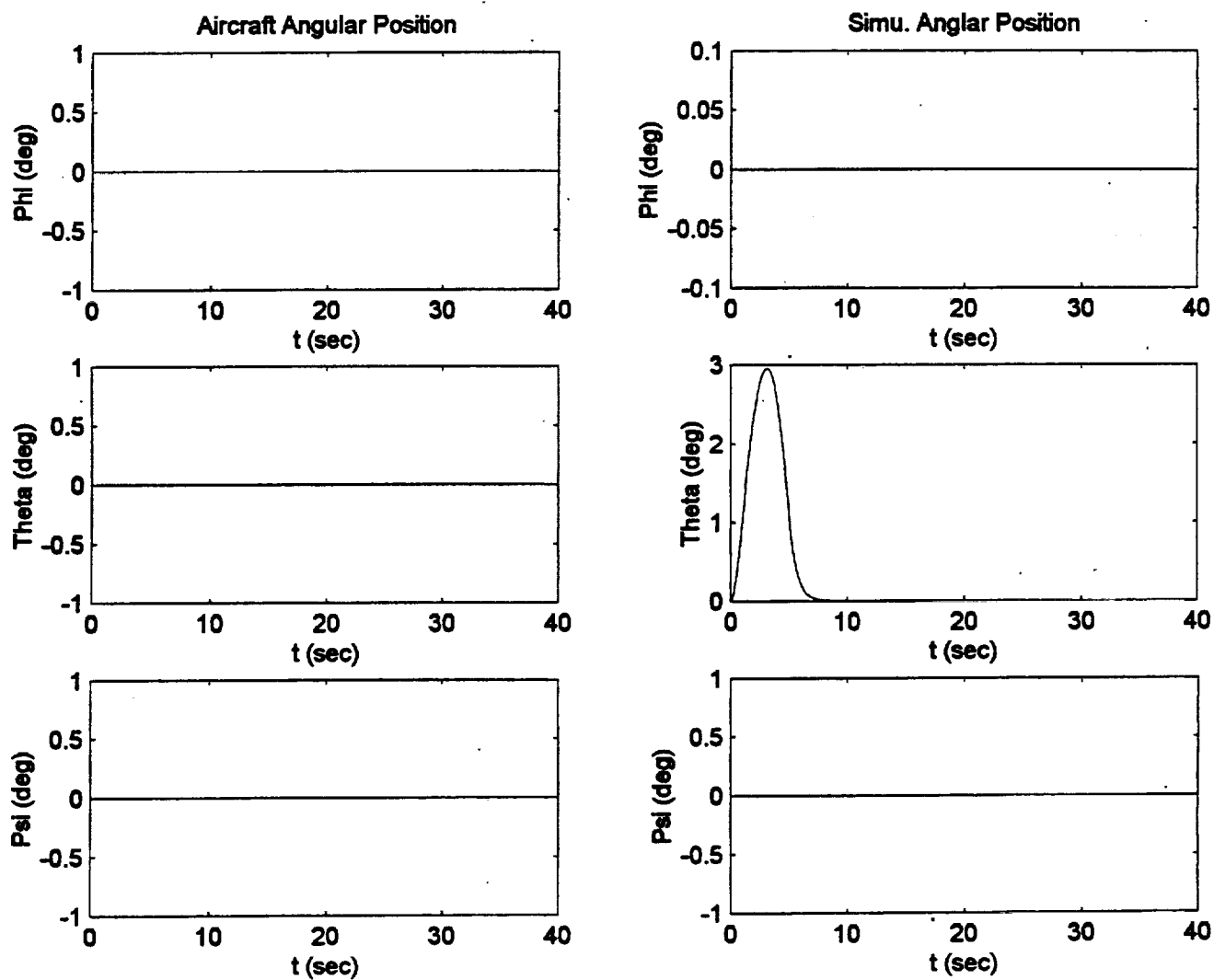


Figure C.2. NASA Adaptive Algorithm Surge Half Sinusoidal Pulse Input.

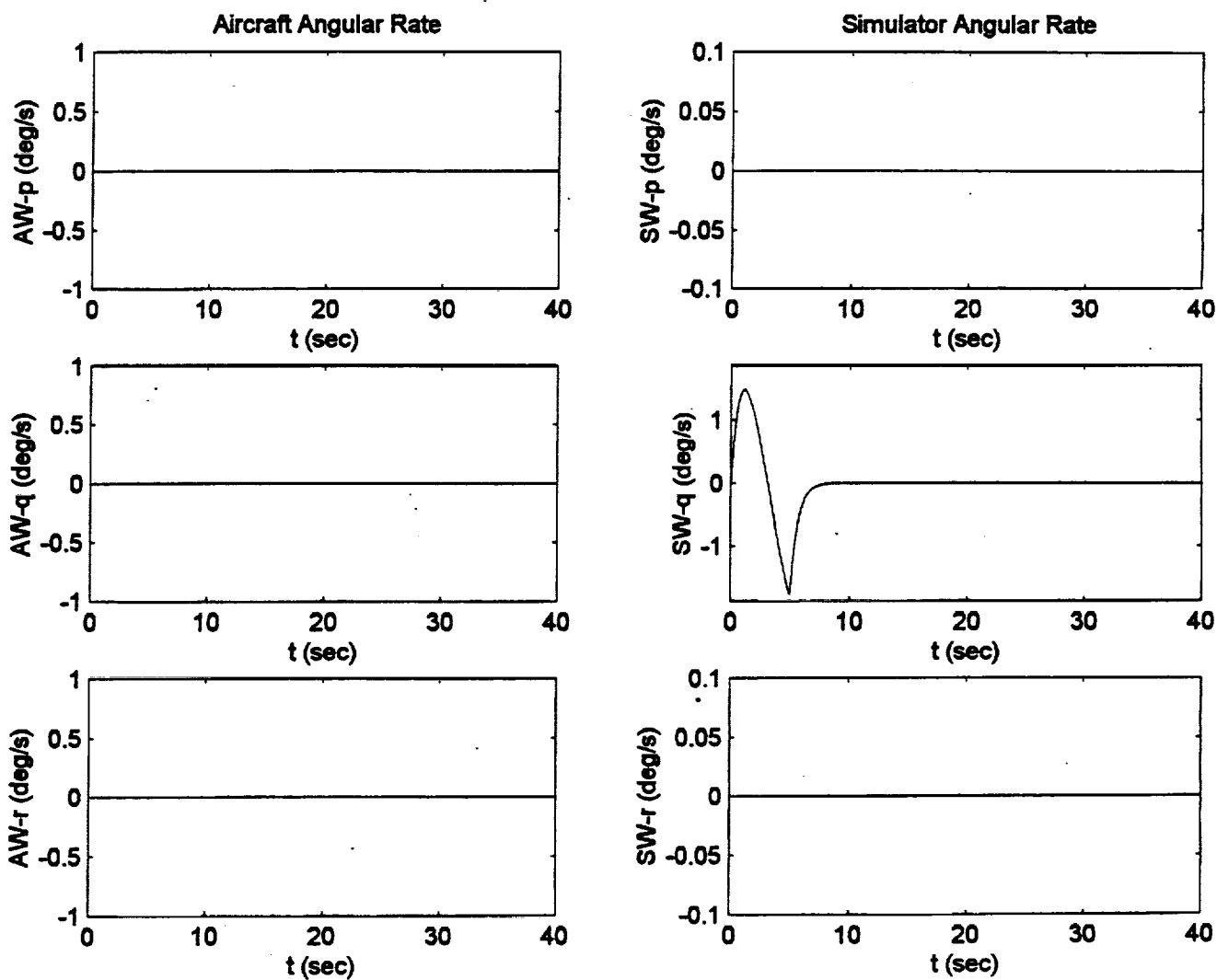


Figure C.2. NASA Adaptive Algorithm Surge Half Sinusoidal Pulse Input.

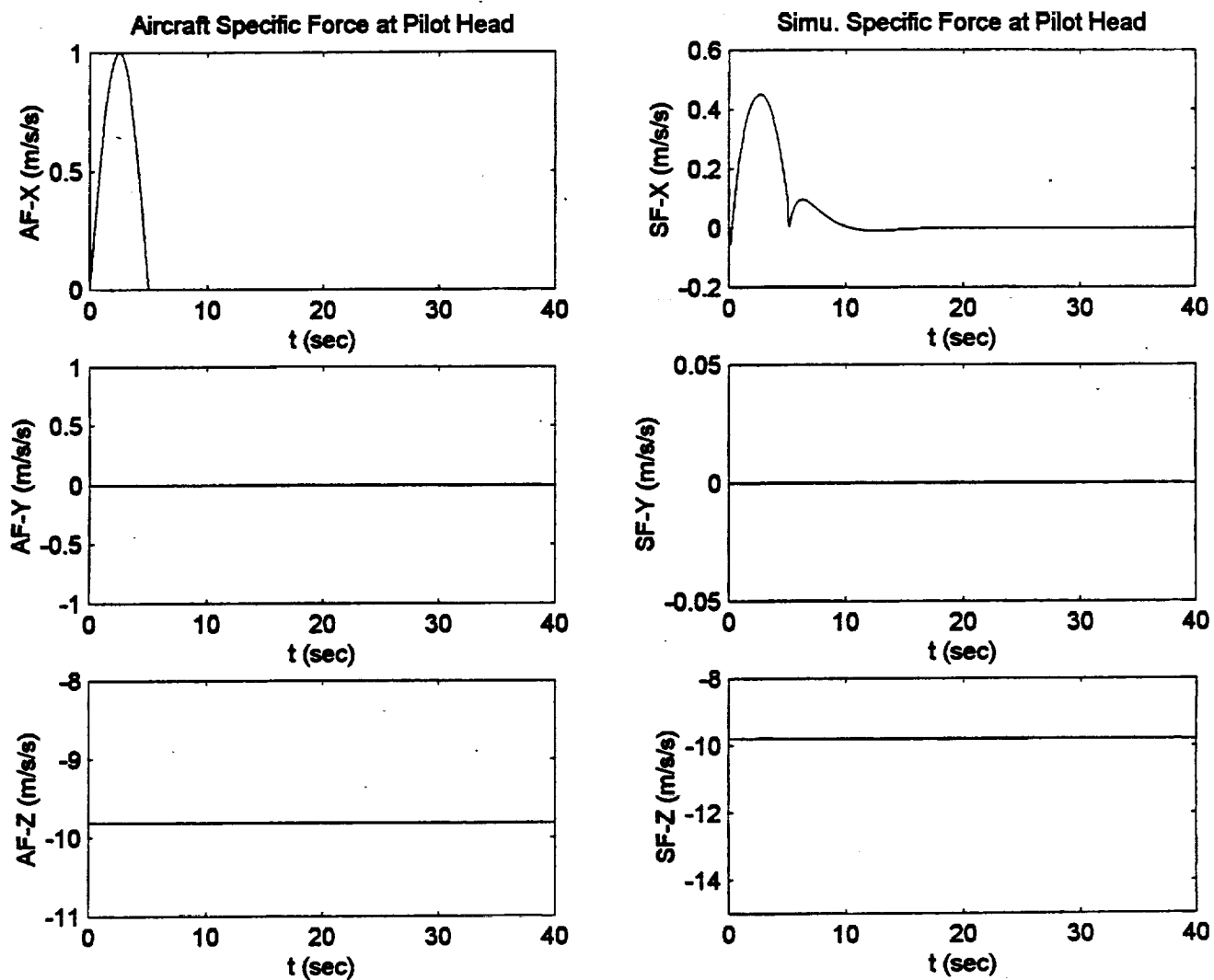


Figure C.2. NASA Adaptive Algorithm Surge Half Sinusoidal Pulse Input.

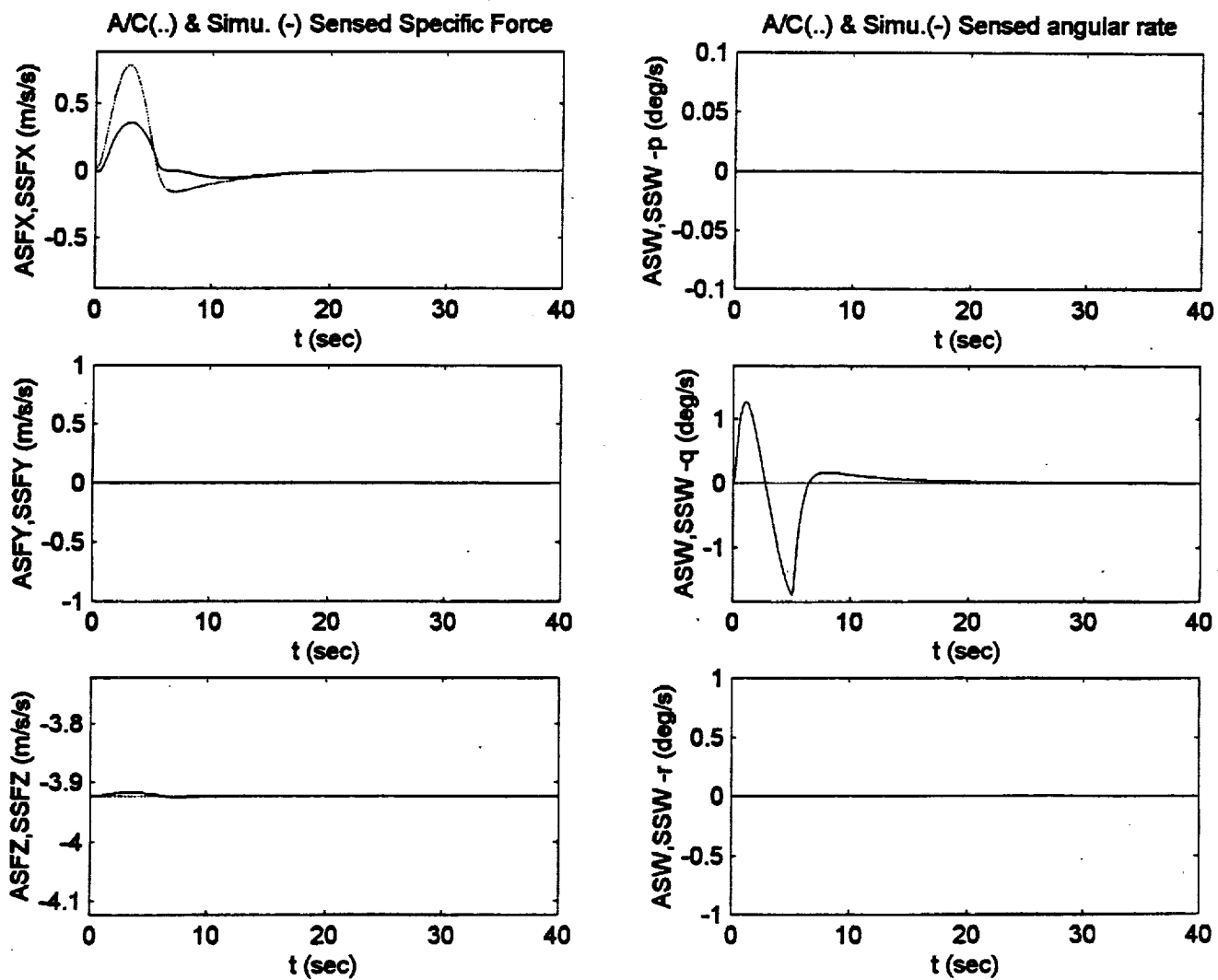


Figure C.2. NASA Adaptive Algorithm Surge Half Sinusoidal Pulse Input.

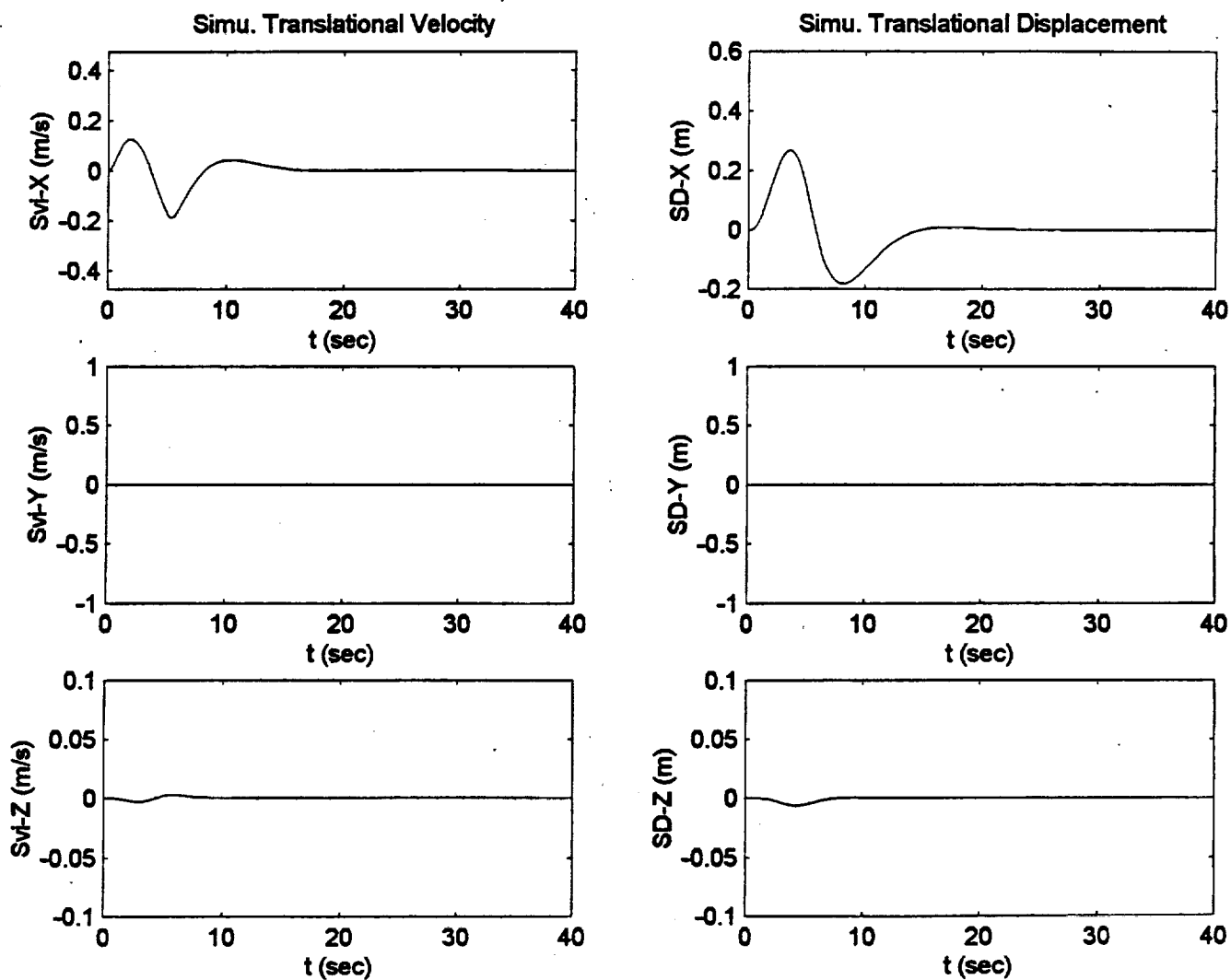


Figure C.2. NASA Adaptive Algorithm Surge Half Sinusoidal Pulse Input.

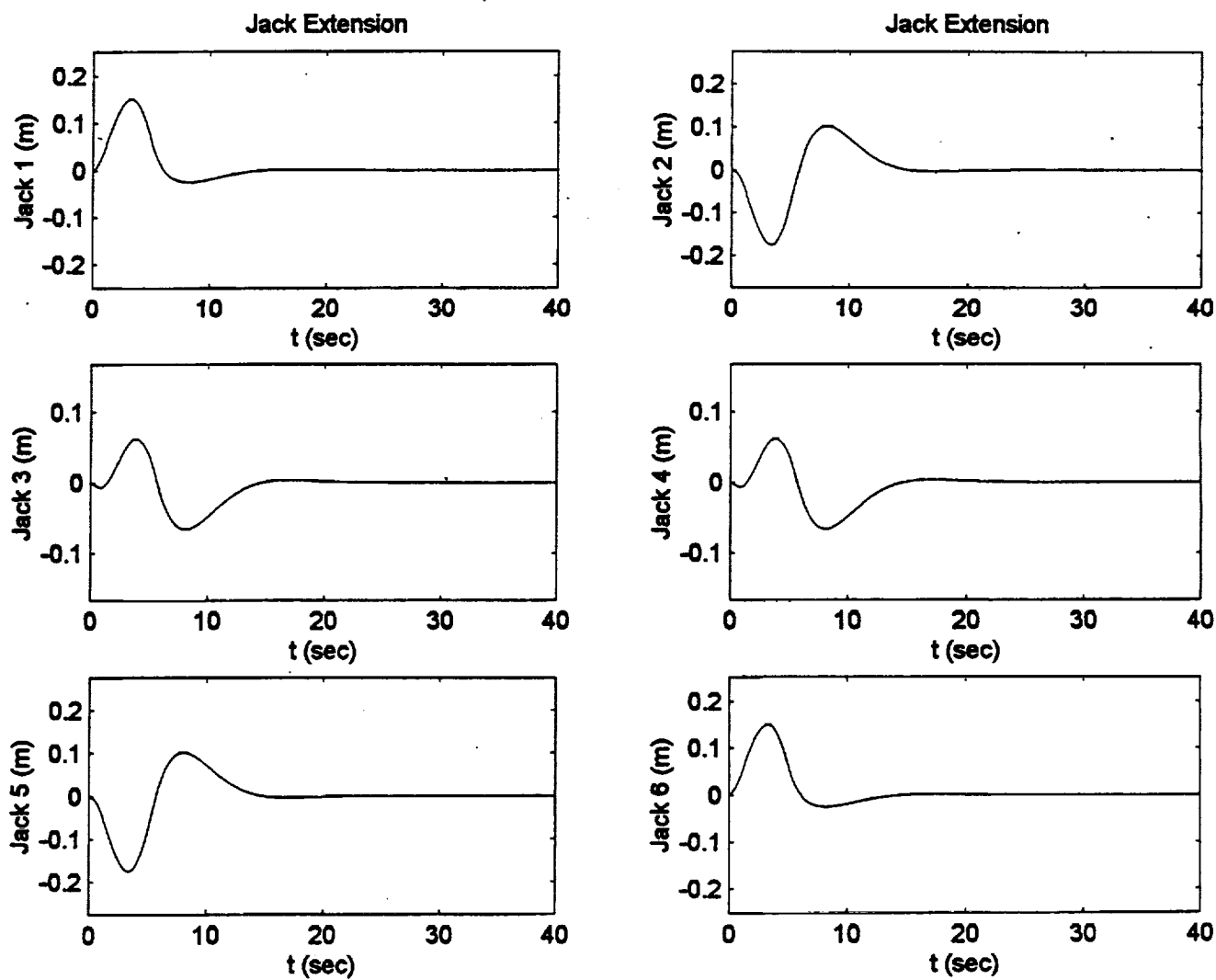


Figure C.2. NASA Adaptive Algorithm Surge Half Sinusoidal Pulse Input.

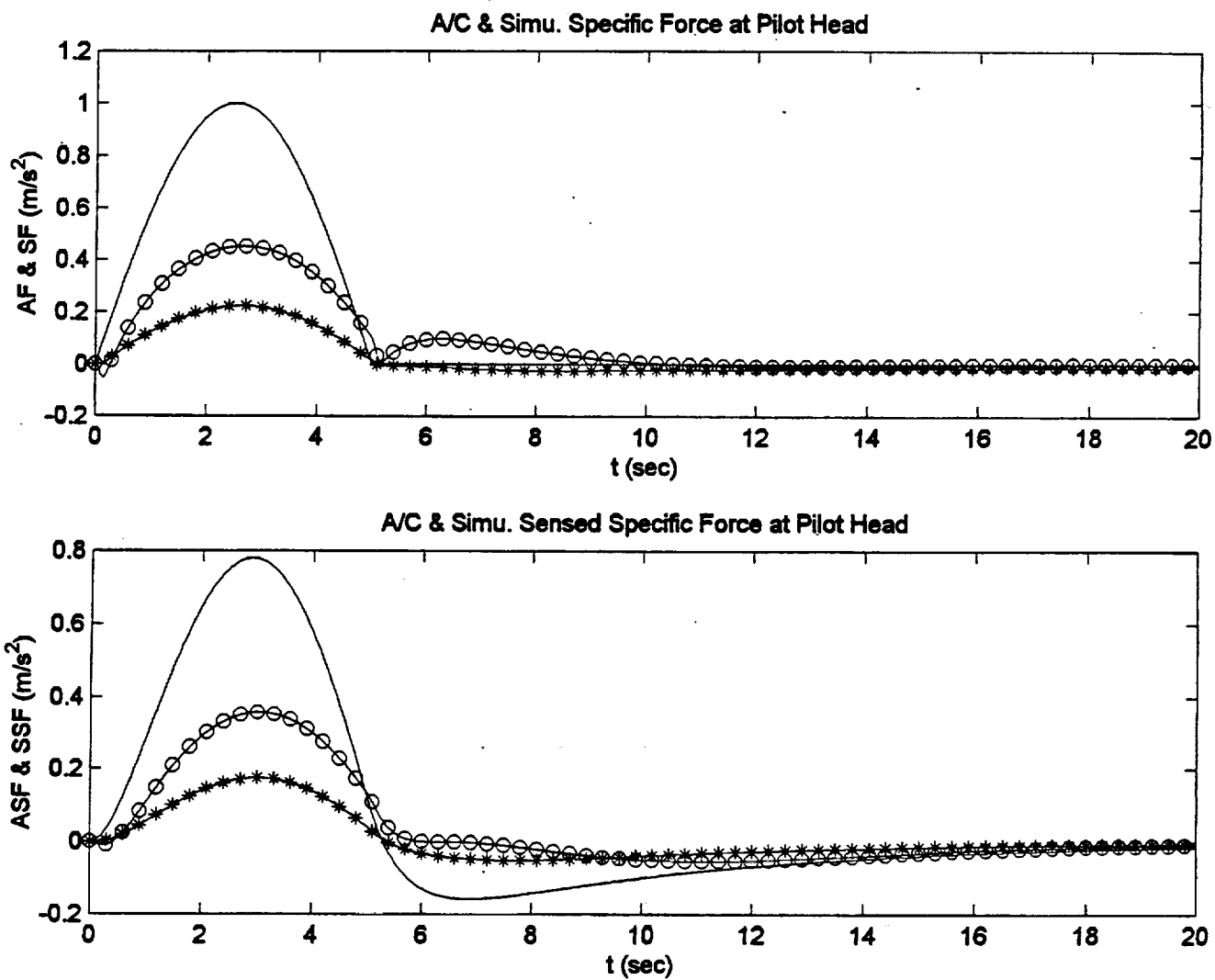


Figure C.3. Surge Pulse Input Specific Force Comparison.

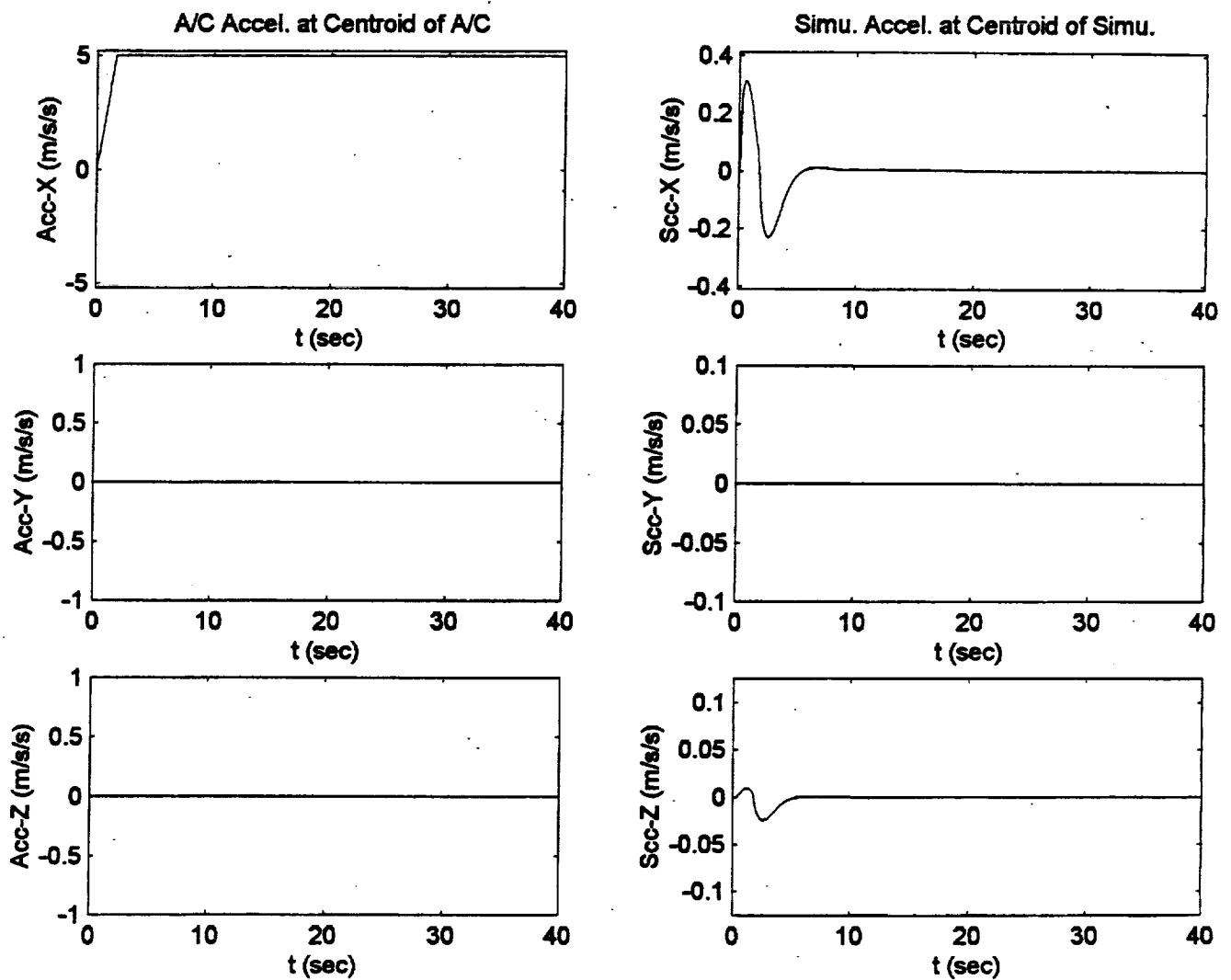


Figure C.4. Optimal Algorithm Surge 3 m/s²/s Ramp-Step Input.

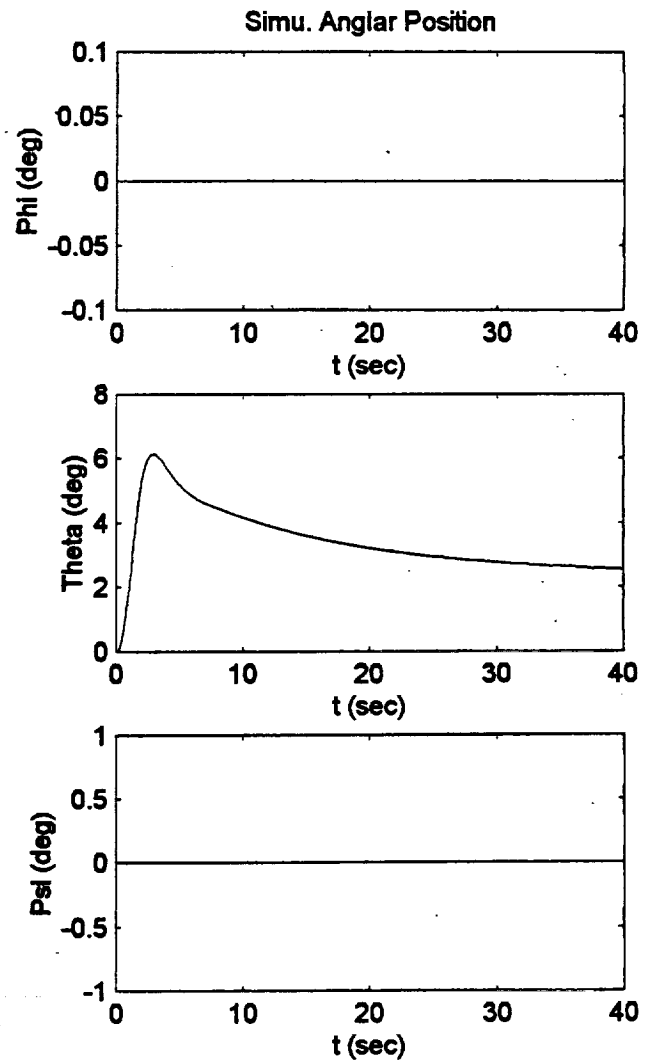
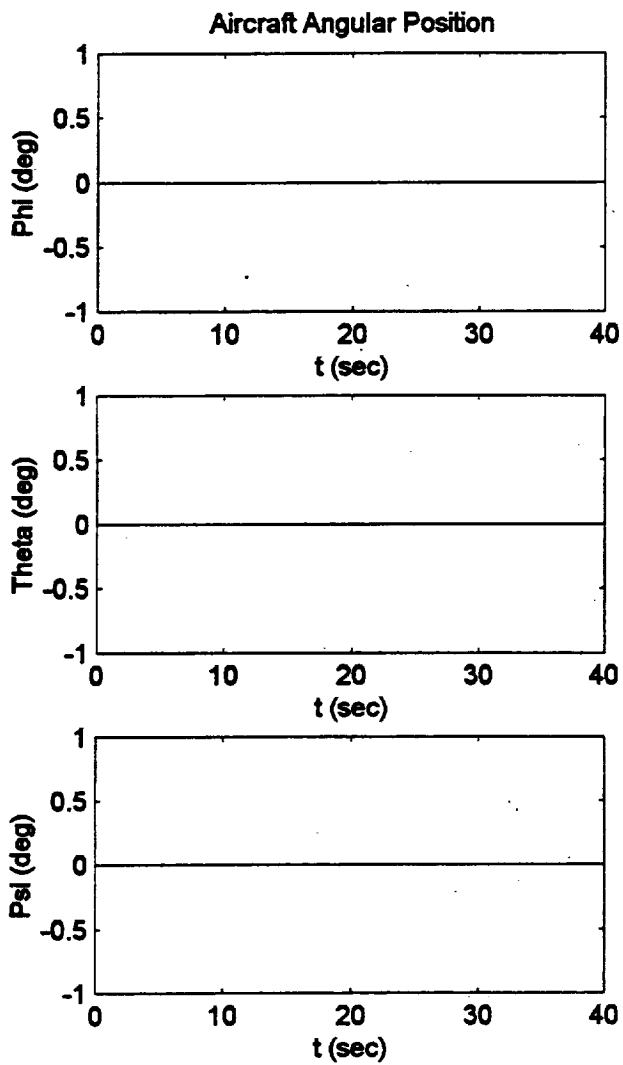


Figure C.4. Optimal Algorithm Surge $3 \text{ m/s}^2/\text{s}$ Ramp-Step Input.

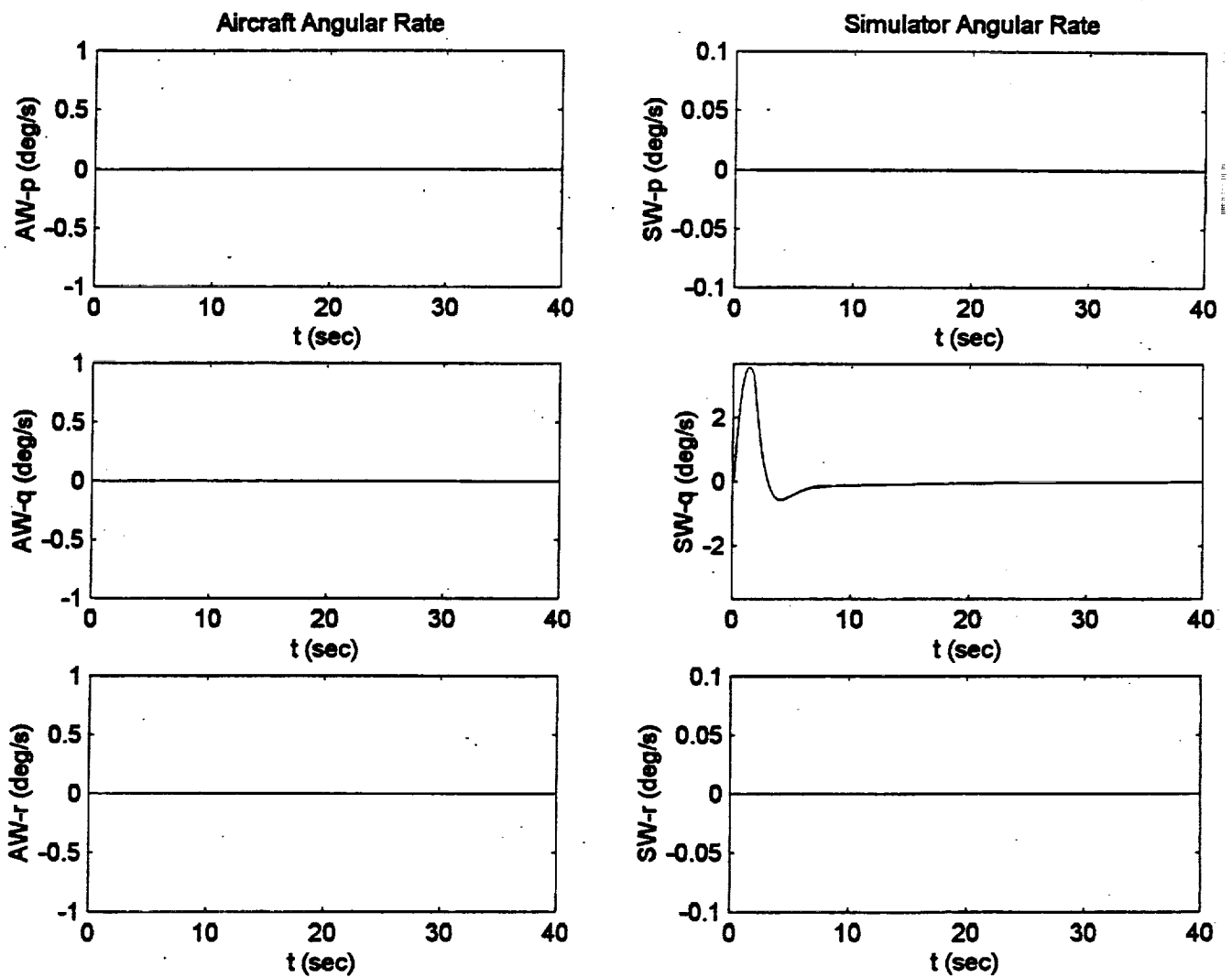


Figure C.4. Optimal Algorithm Surge $3 \text{ m/s}^2/\text{s}$ Ramp-Step Input.

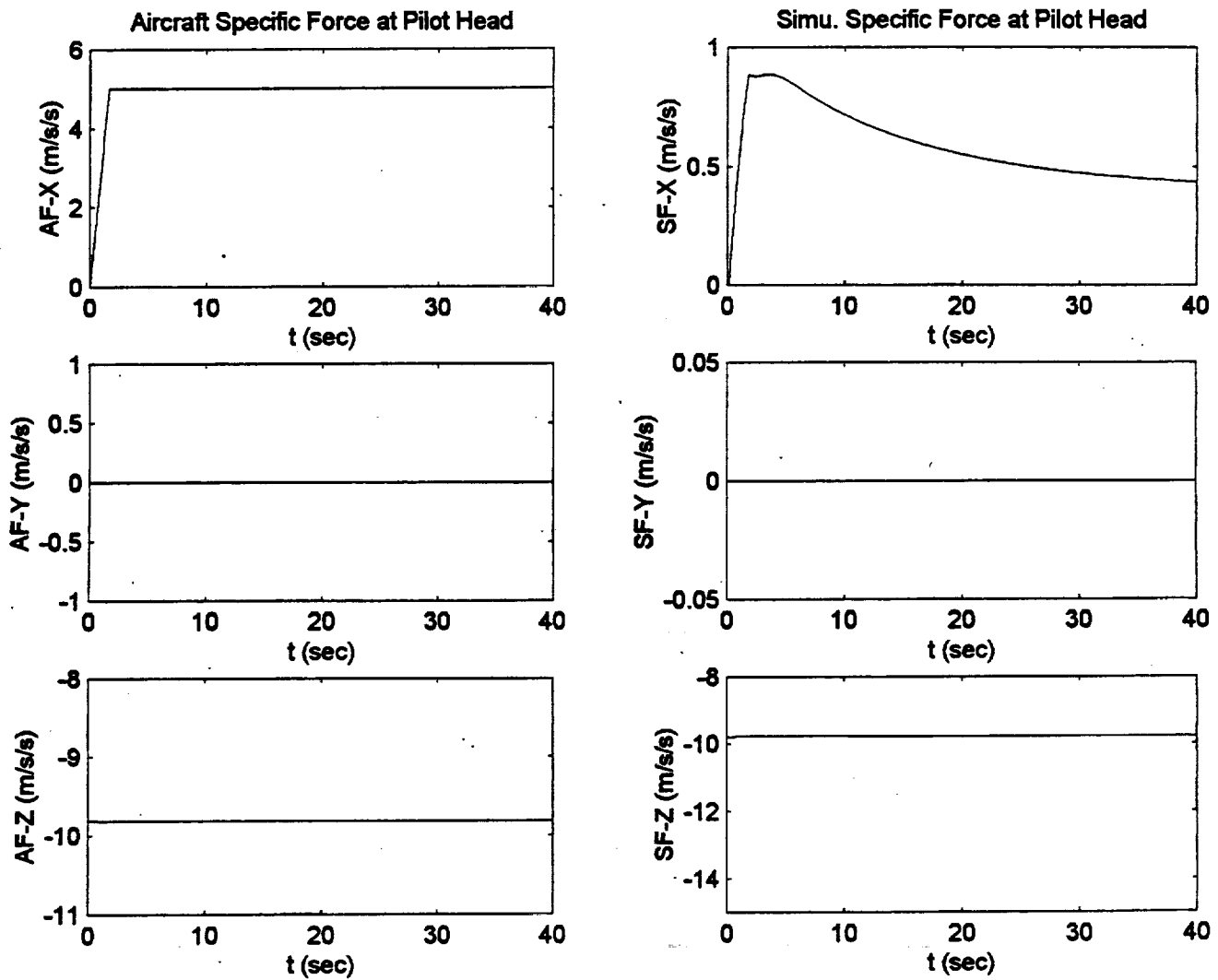


Figure C.4. Optimal Algorithm Surge $3 \text{ m/s}^2/\text{s}$ Ramp-Step Input.

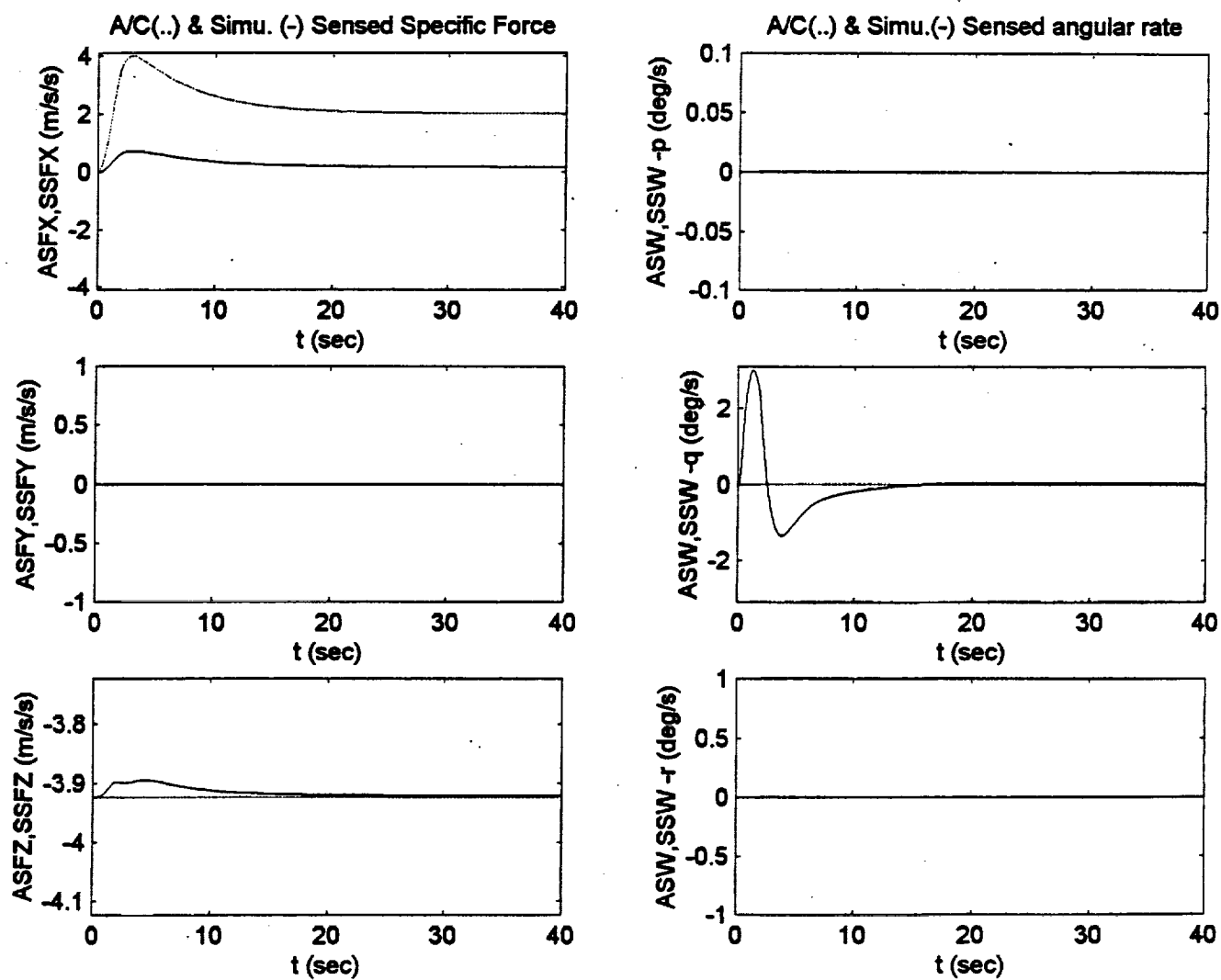


Figure C.4. Optimal Algorithm Surge $3 \text{ m/s}^2/\text{s}$ Ramp-Step Input.

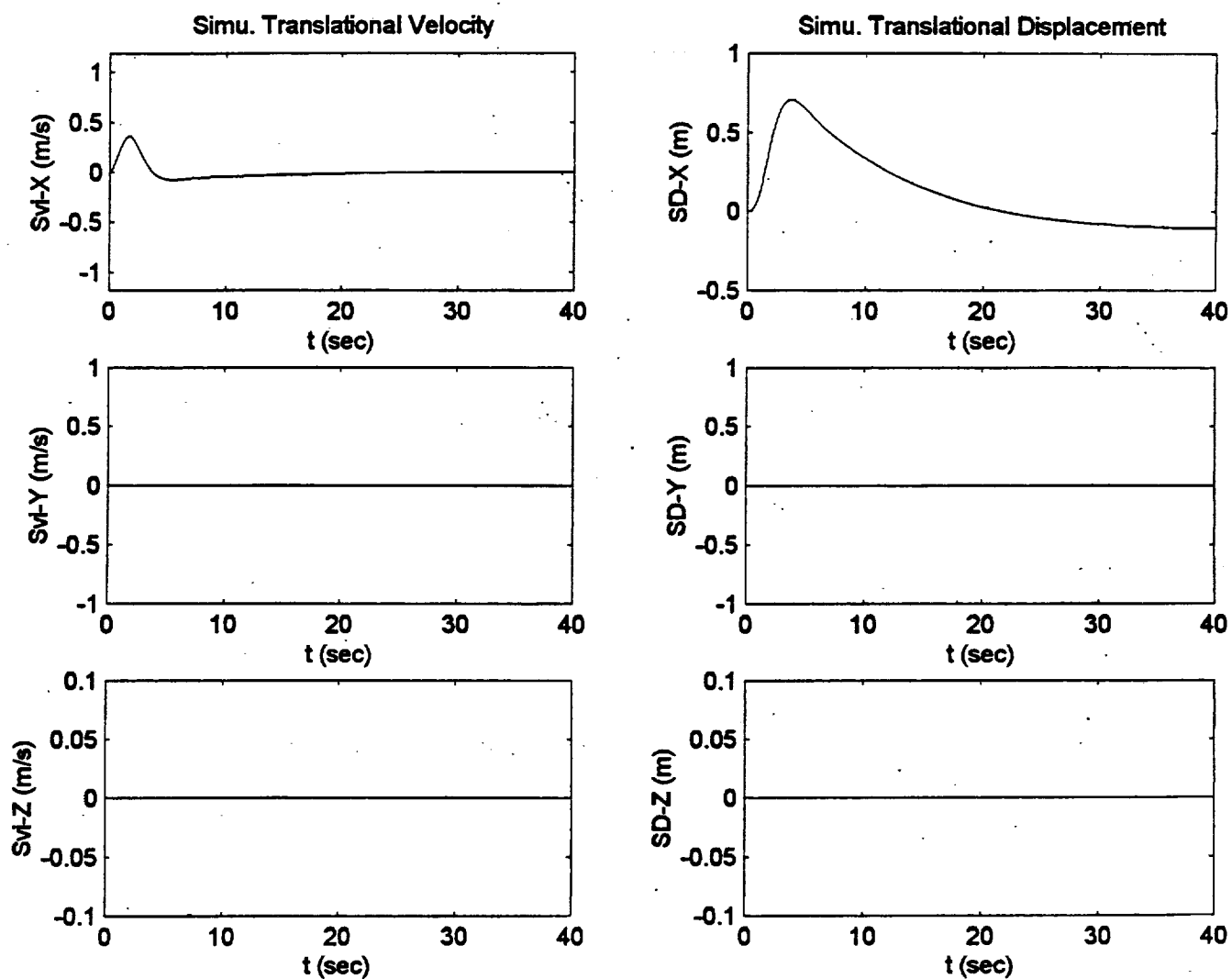


Figure C.4. Optimal Algorithm Surge $3 \text{ m/s}^2/\text{s}$ Ramp-Step Input.

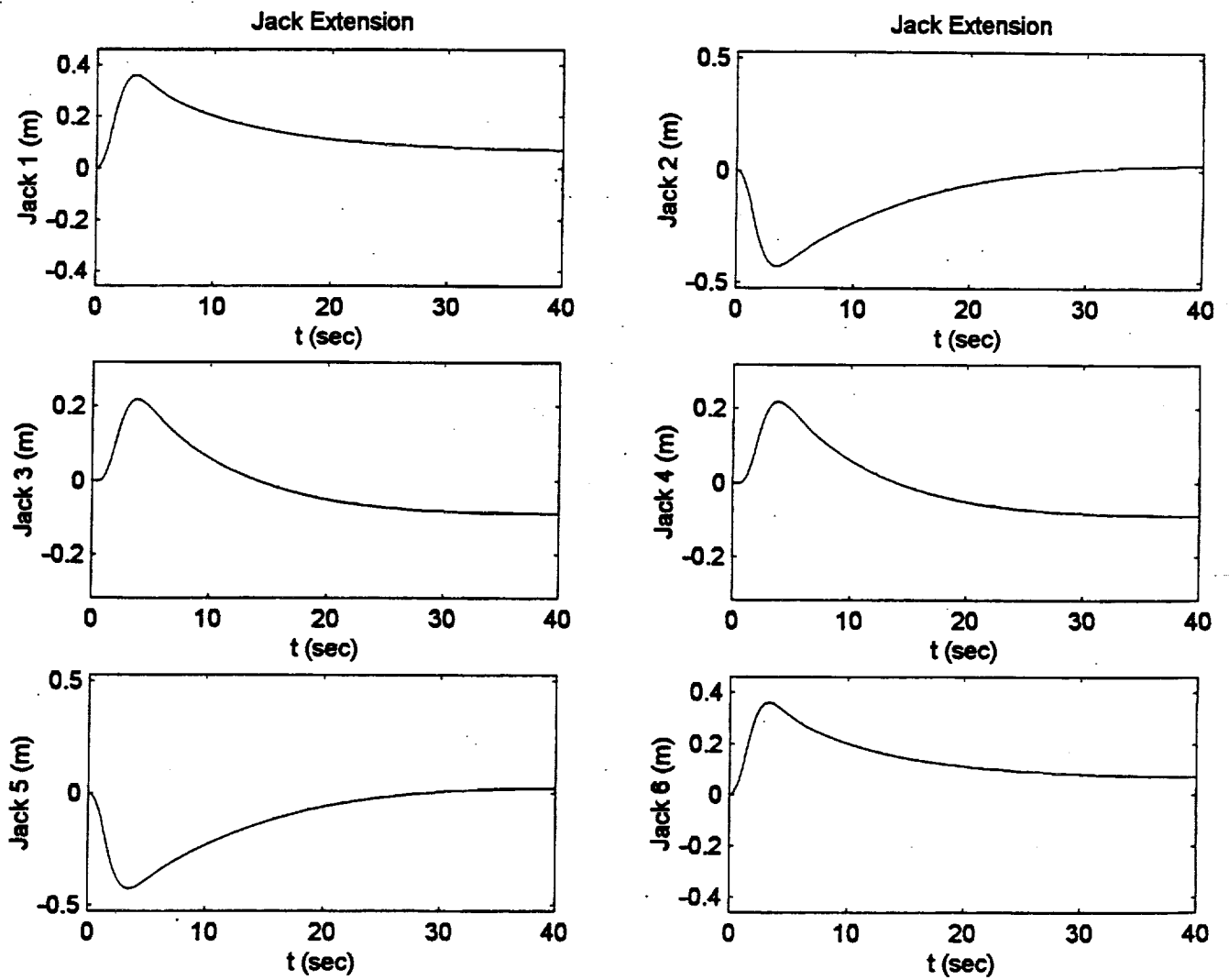


Figure C.4. Optimal Algorithm Surge $3 \text{ m/s}^2/\text{s}$ Ramp-Step Input.

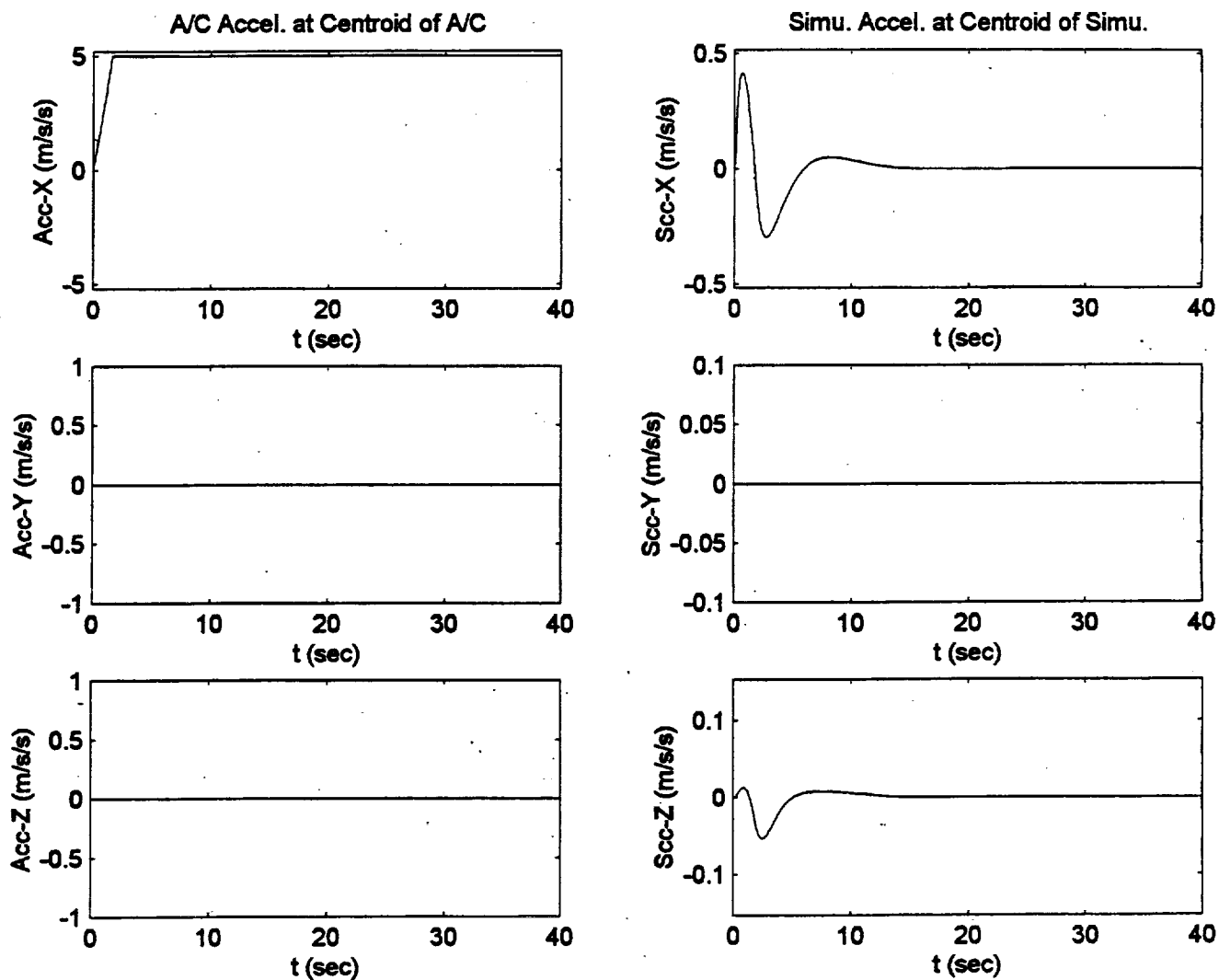


Figure C.5. NASA Adaptive Algorithm Surge $3 \text{ m/s}^2/\text{s}$ Ramp-Step Input.

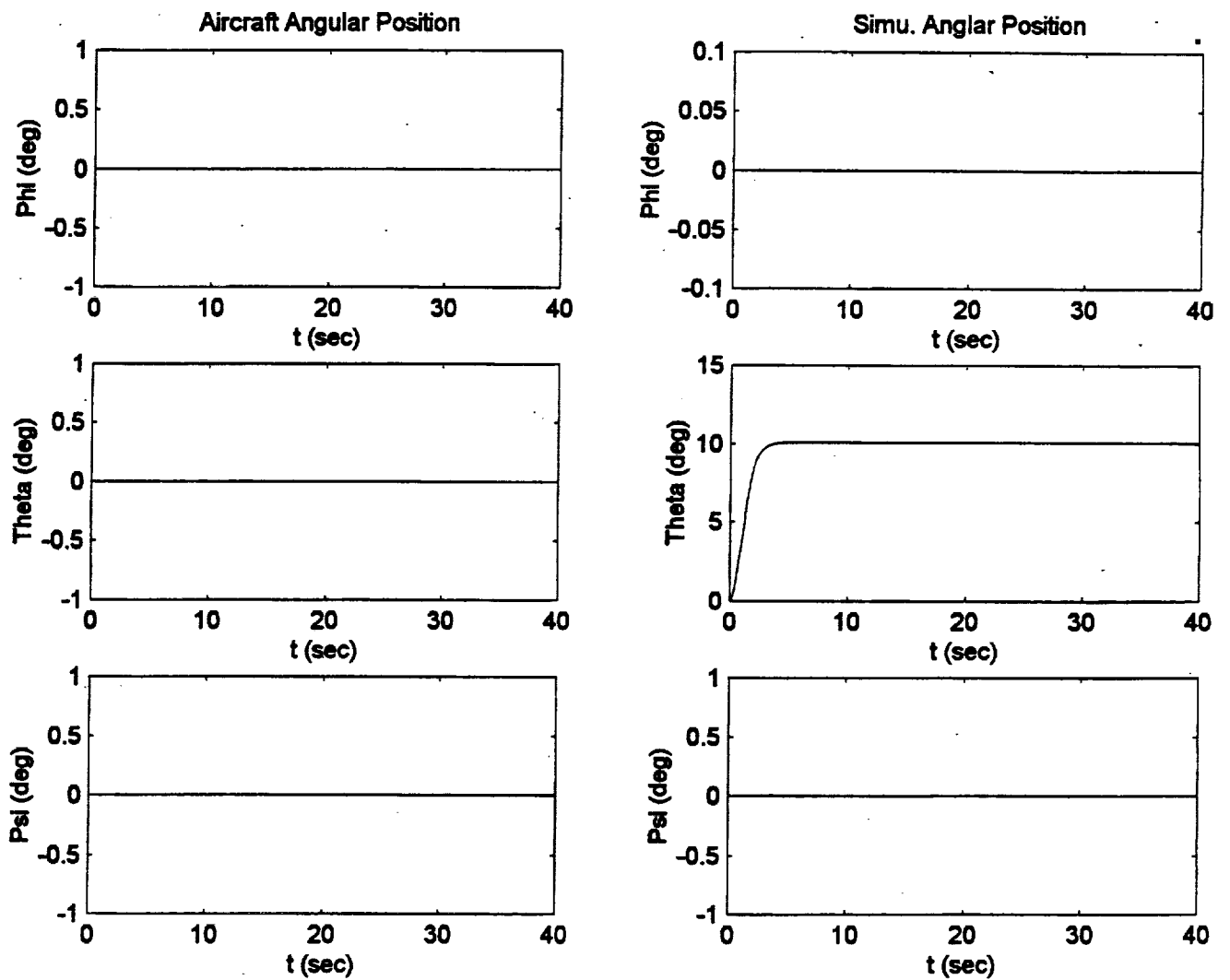


Figure C.5. NASA Adaptive Algorithm Surge $3 \text{ m/s}^2/\text{s}$ Ramp-Step Input.

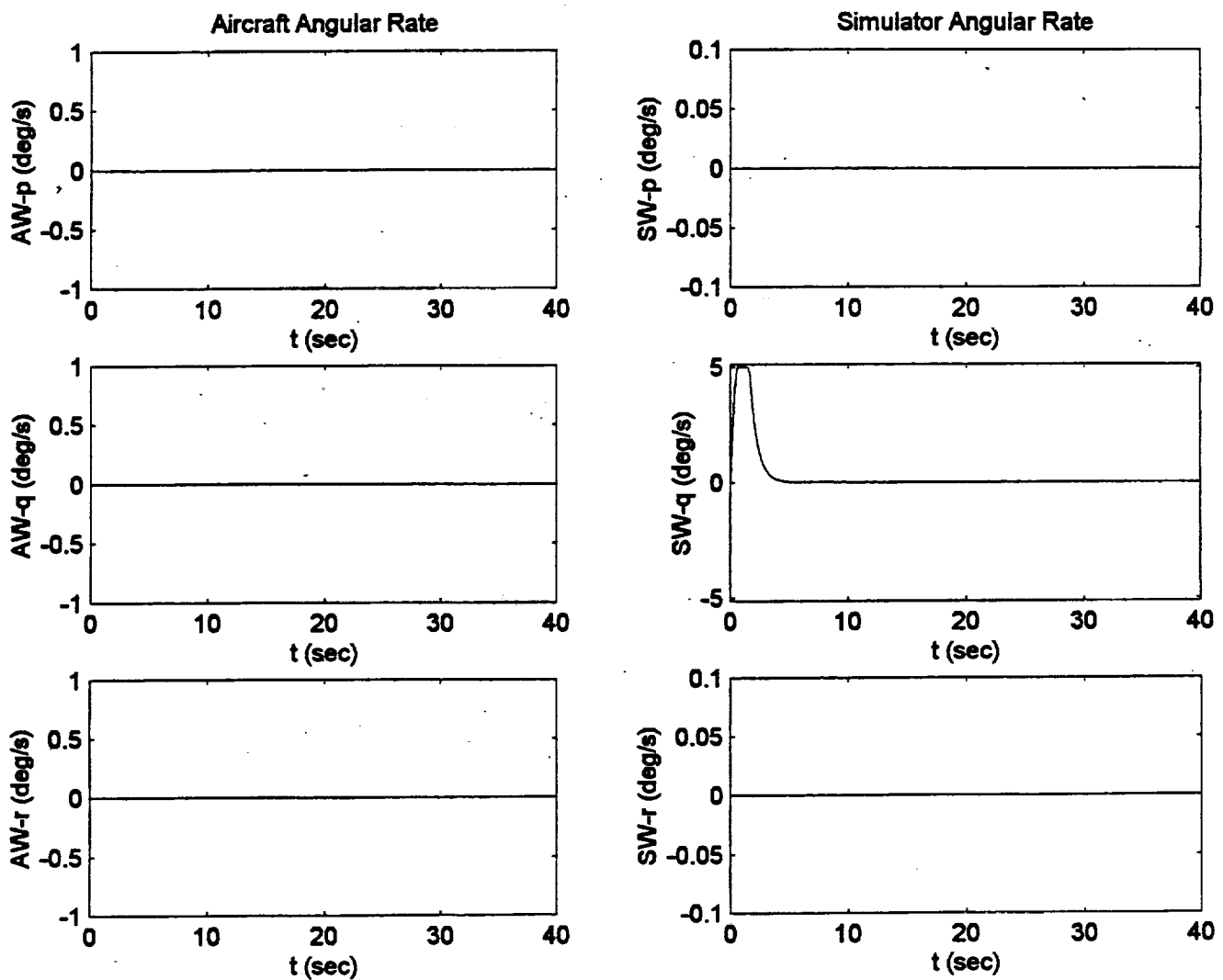


Figure C.5. NASA Adaptive Algorithm Surge $3 \text{ m/s}^2/\text{s}$ Ramp-Step Input.

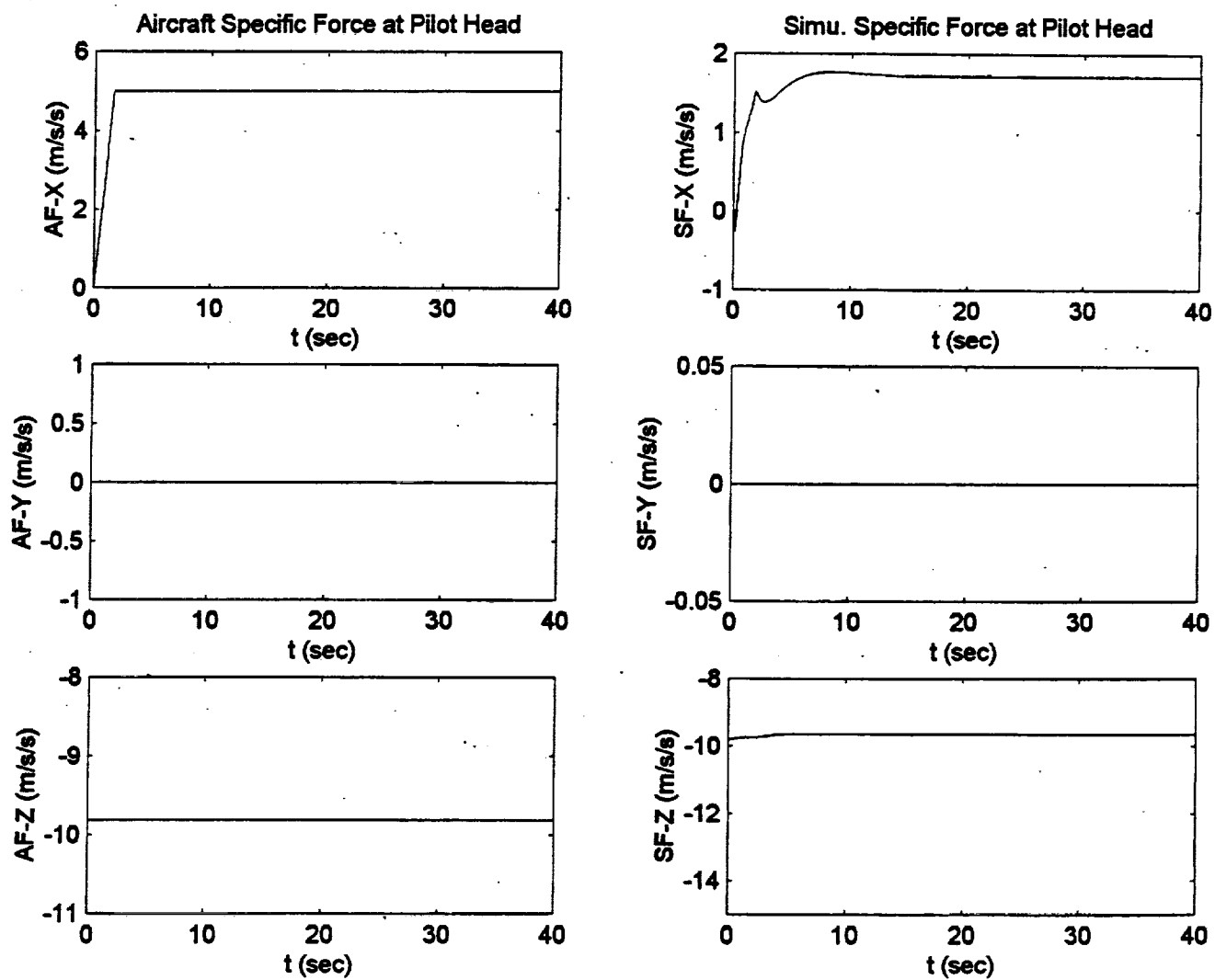


Figure C.5. NASA Adaptive Algorithm Surge 3 m/s²/s Ramp-Step Input.

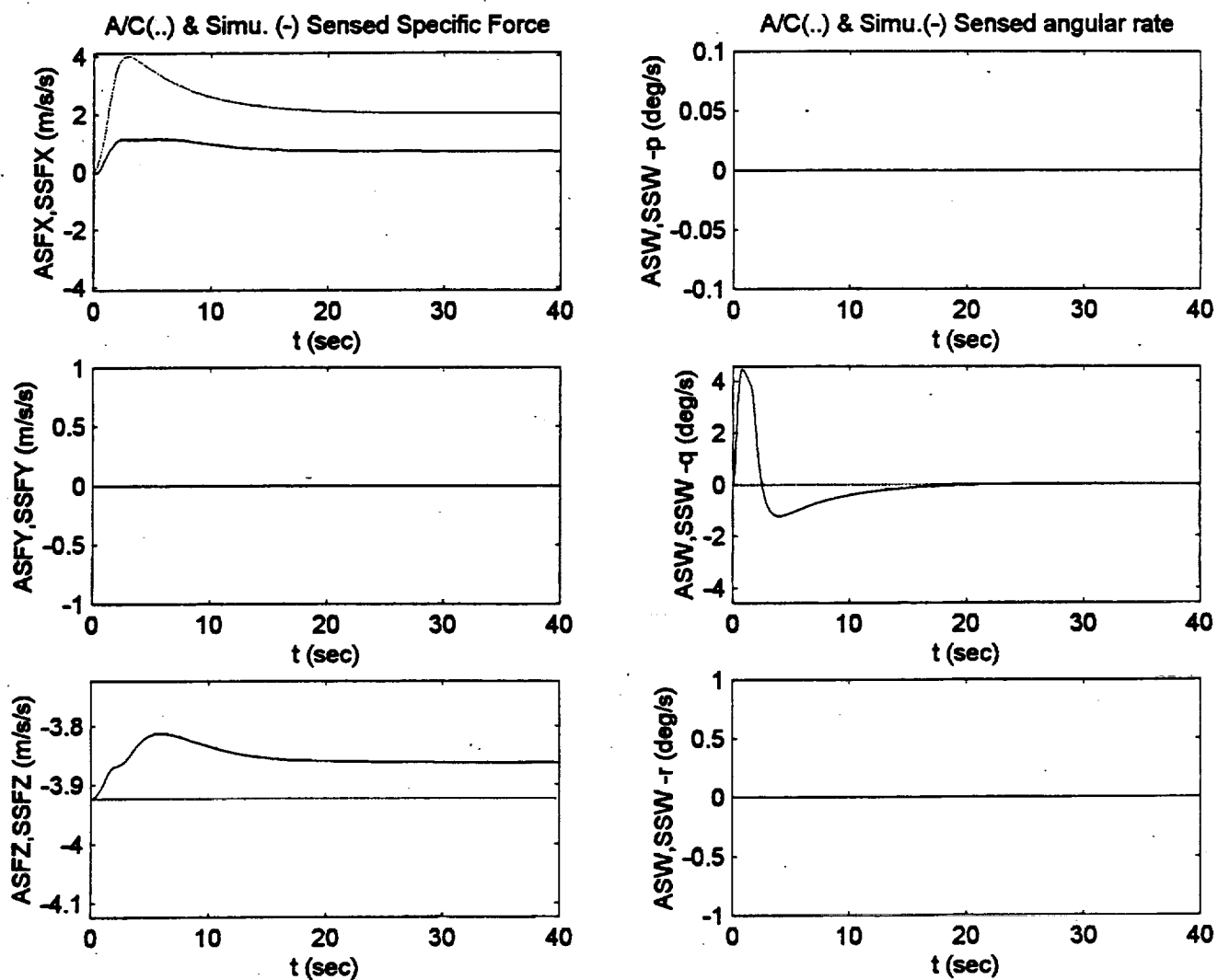


Figure C.5. NASA Adaptive Algorithm Surge 3 m/s²/s Ramp-Step Input.

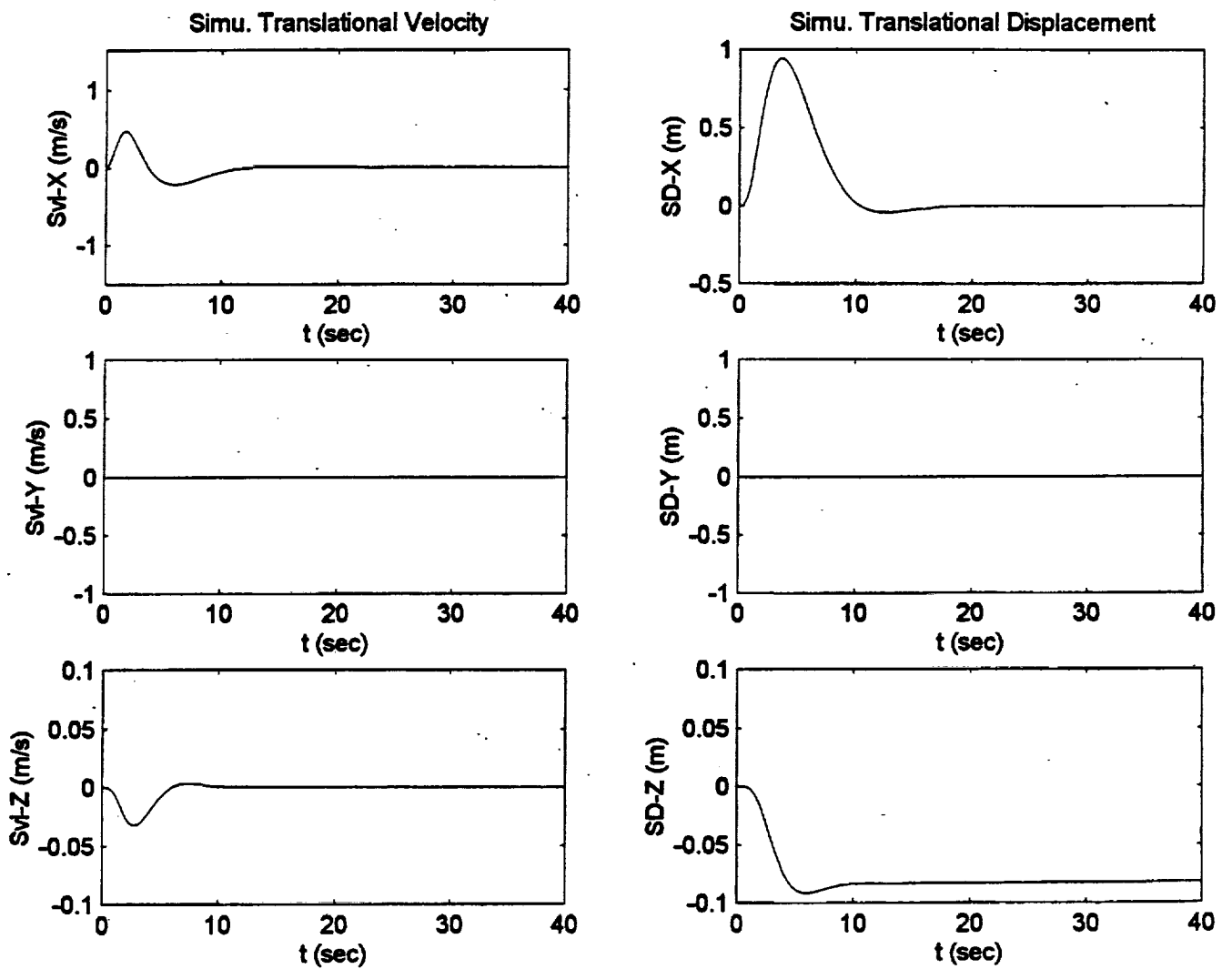


Figure C.5. NASA Adaptive Algorithm Surge 3 m/s²/s Ramp-Step Input.

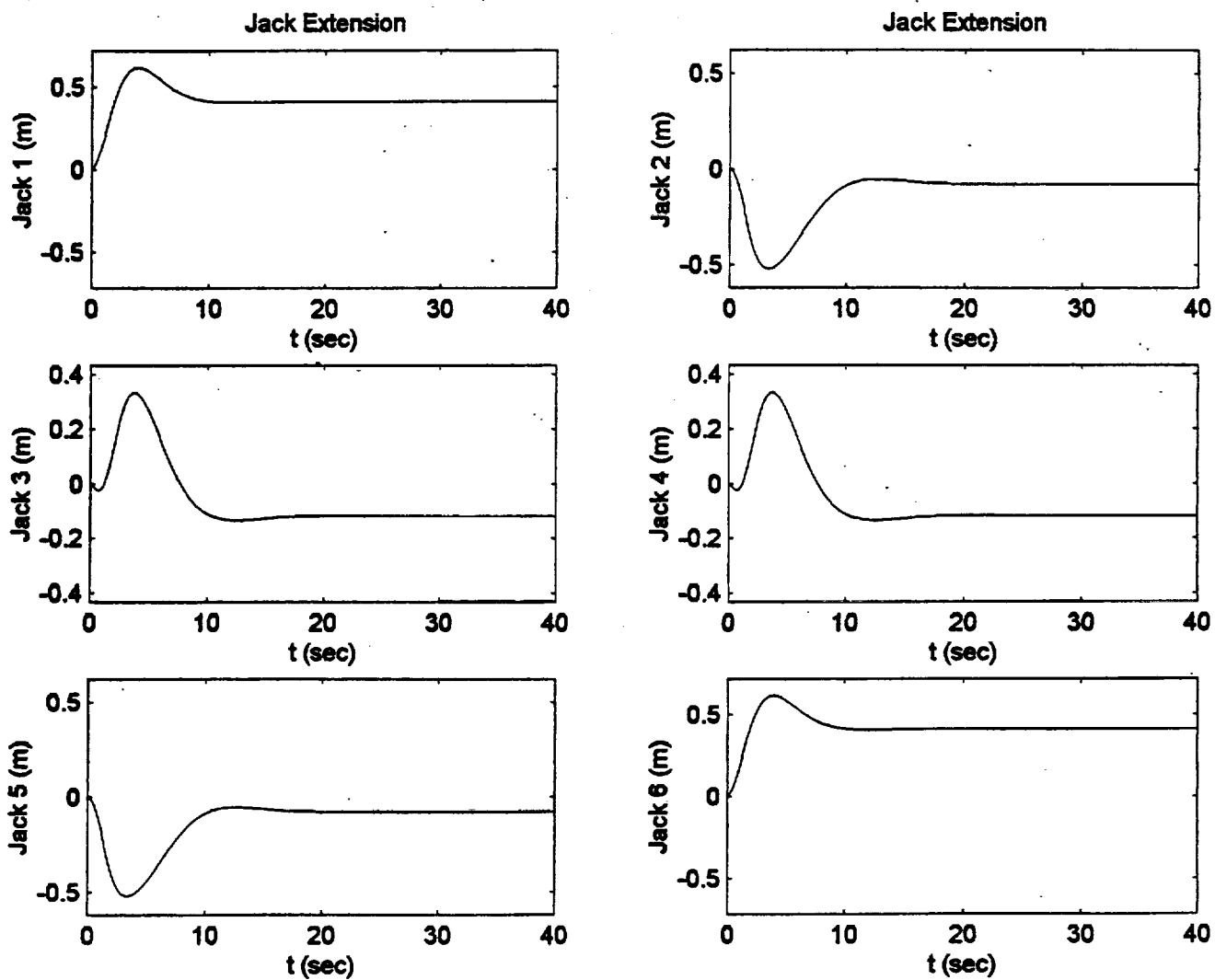


Figure C.5. NASA Adaptive Algorithm Surge $3 \text{ m/s}^2/\text{s}$ Ramp-Step Input.

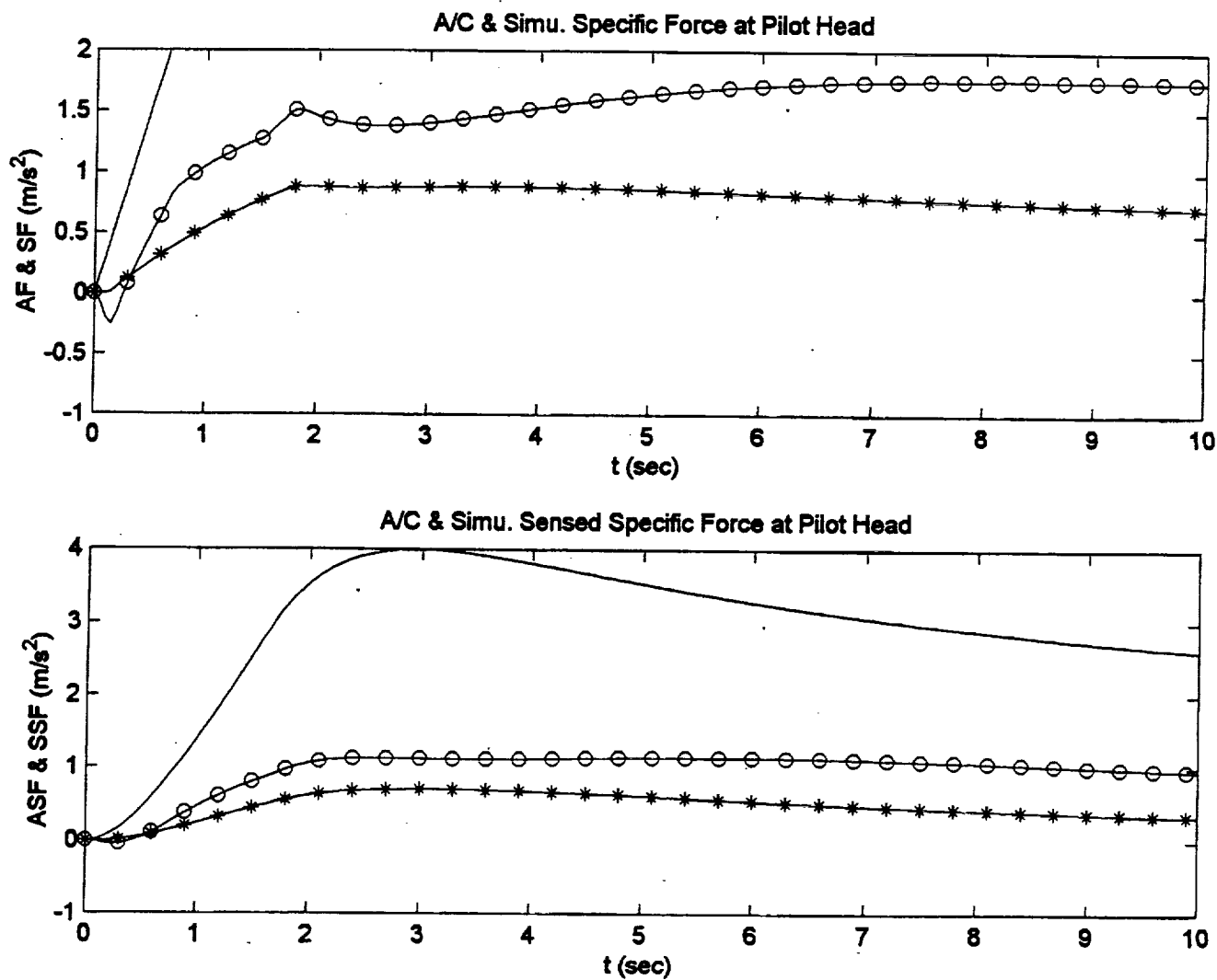


Figure C.6. Surge 3 $m/s^2/s$ Ramp-Step Input Specific Force Comparison.

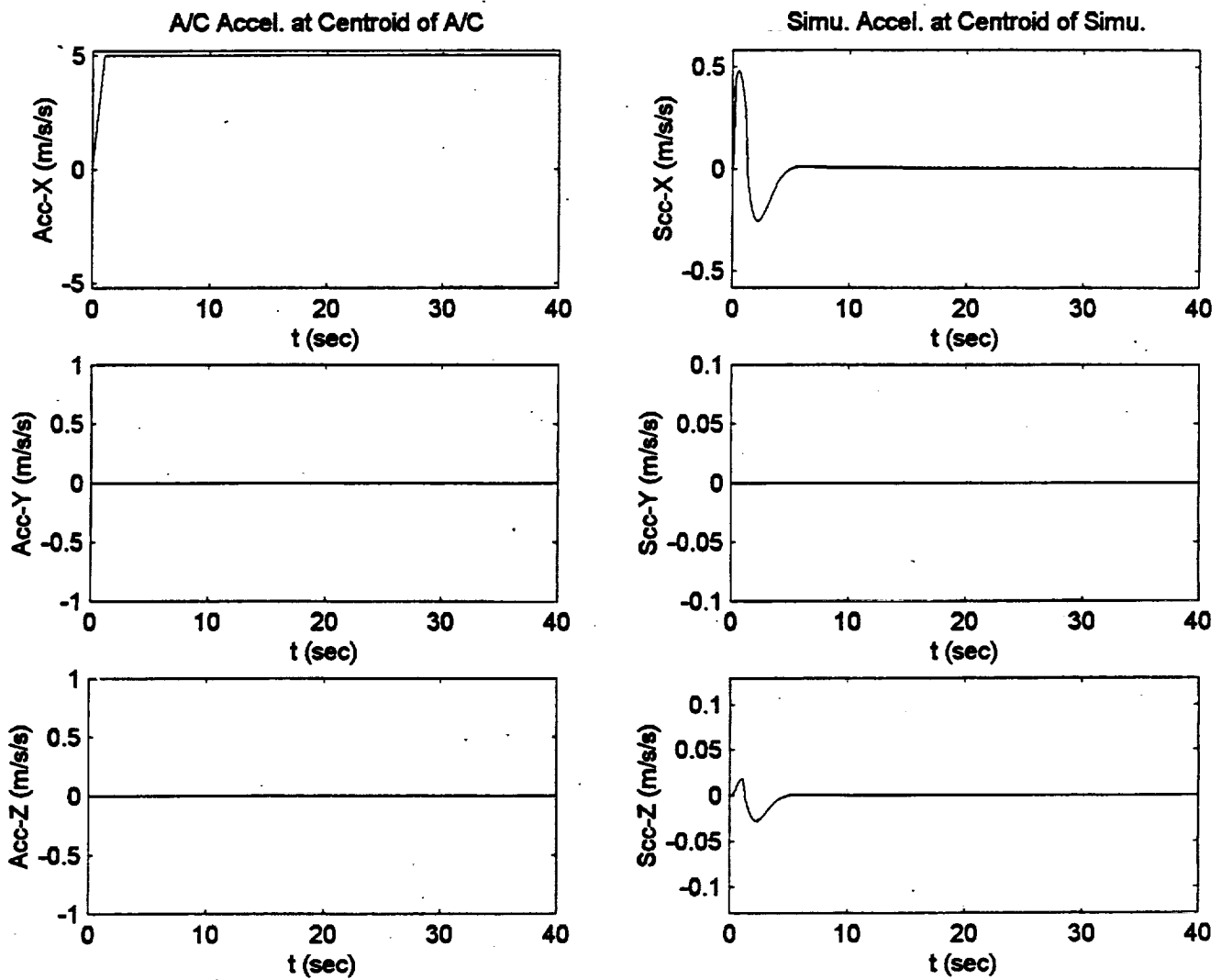


Figure C.7. Optimal Algorithm Surge 5 m/s²/s Ramp-Step Input.

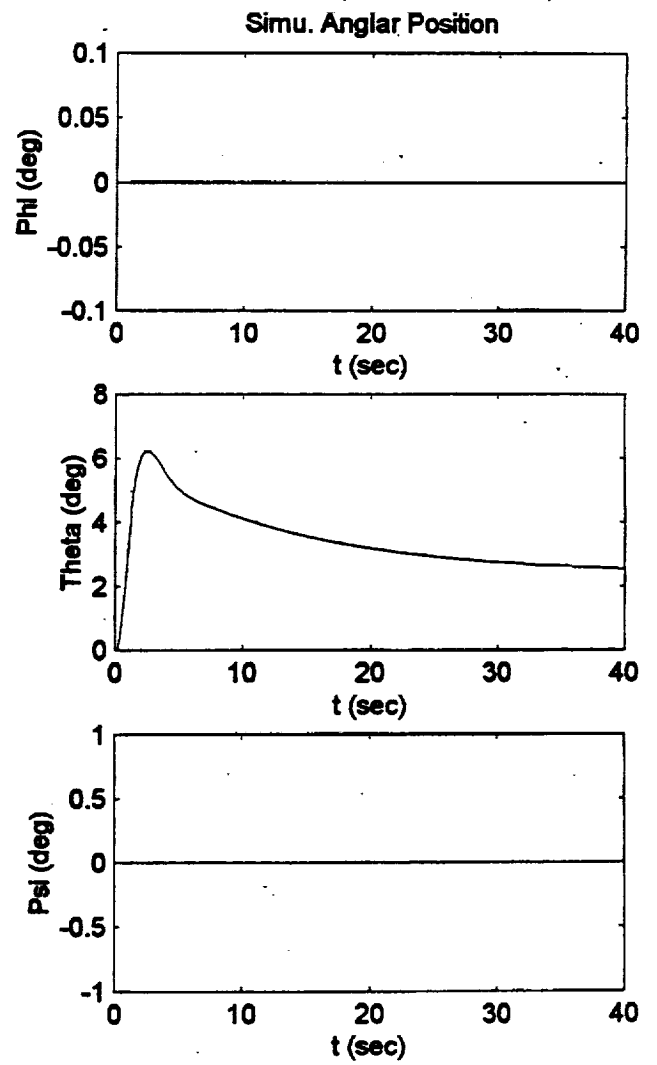
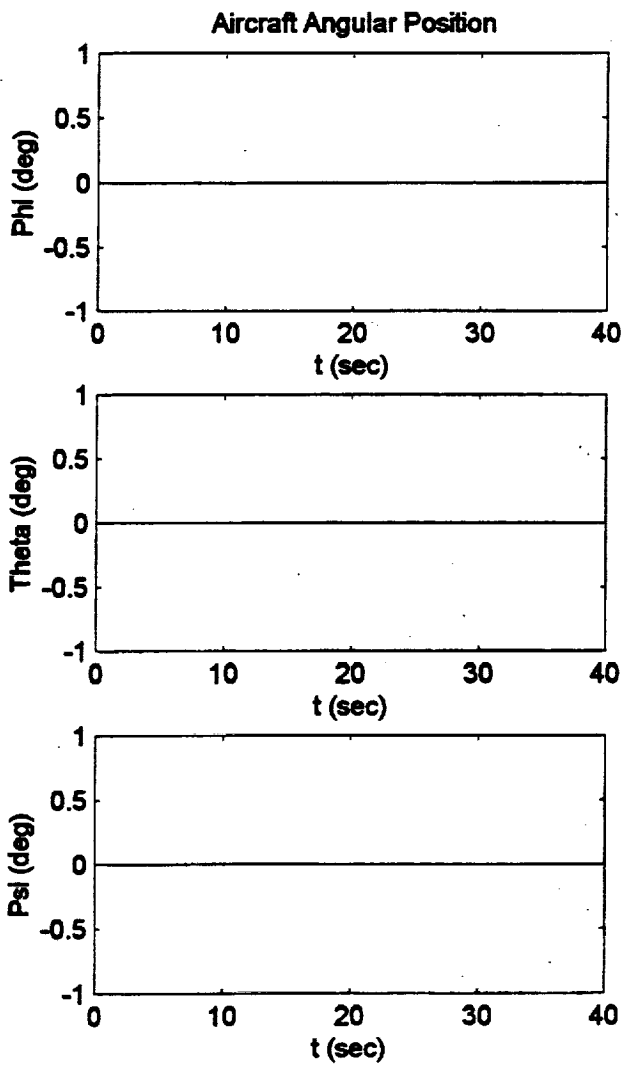


Figure C.7. Optimal Algorithm Surge $5 \text{ m/s}^2/\text{s}$ Ramp-Step Input.

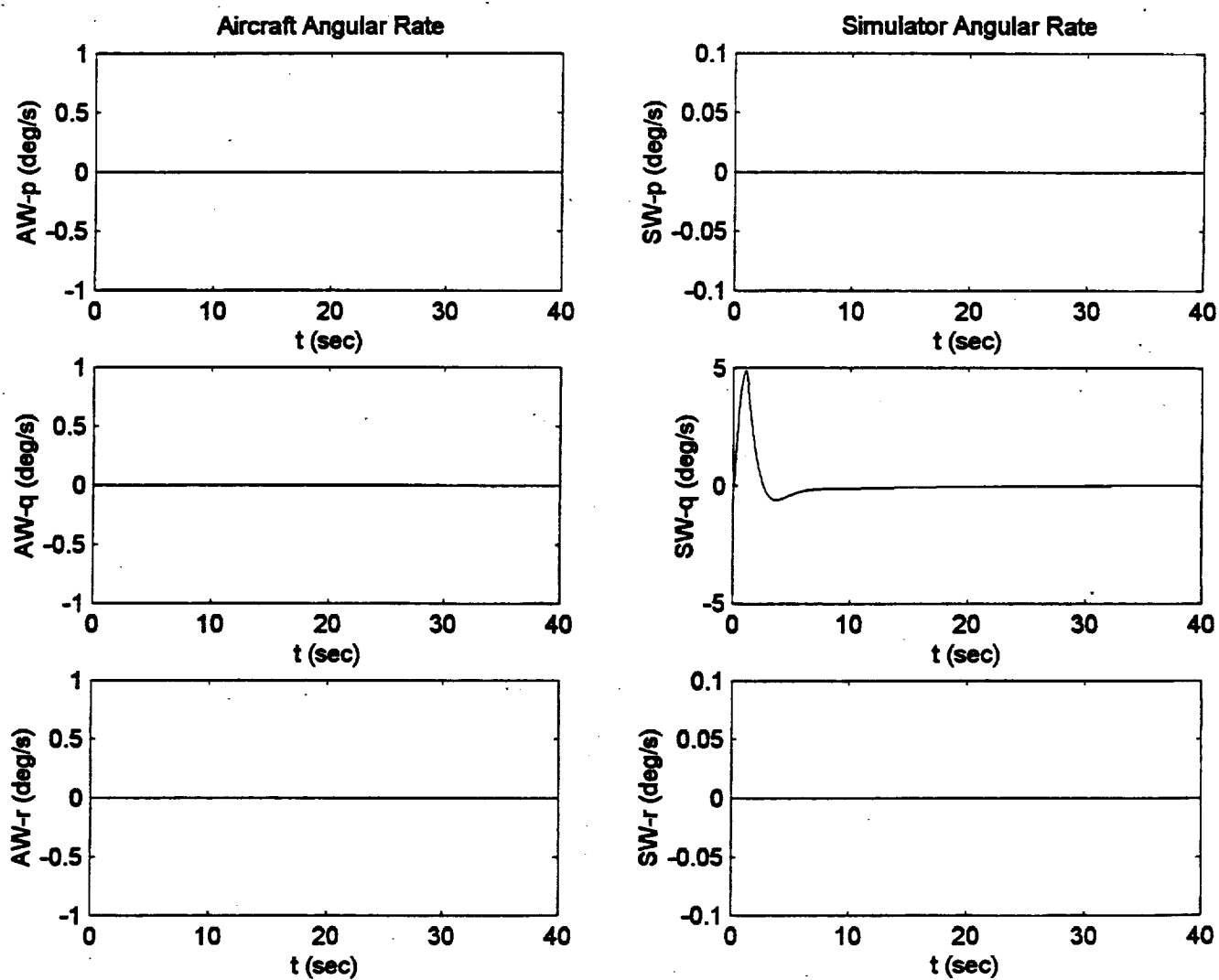


Figure C.7. Optimal Algorithm Surge $5 \text{ m/s}^2/\text{s}$ Ramp-Step Input.

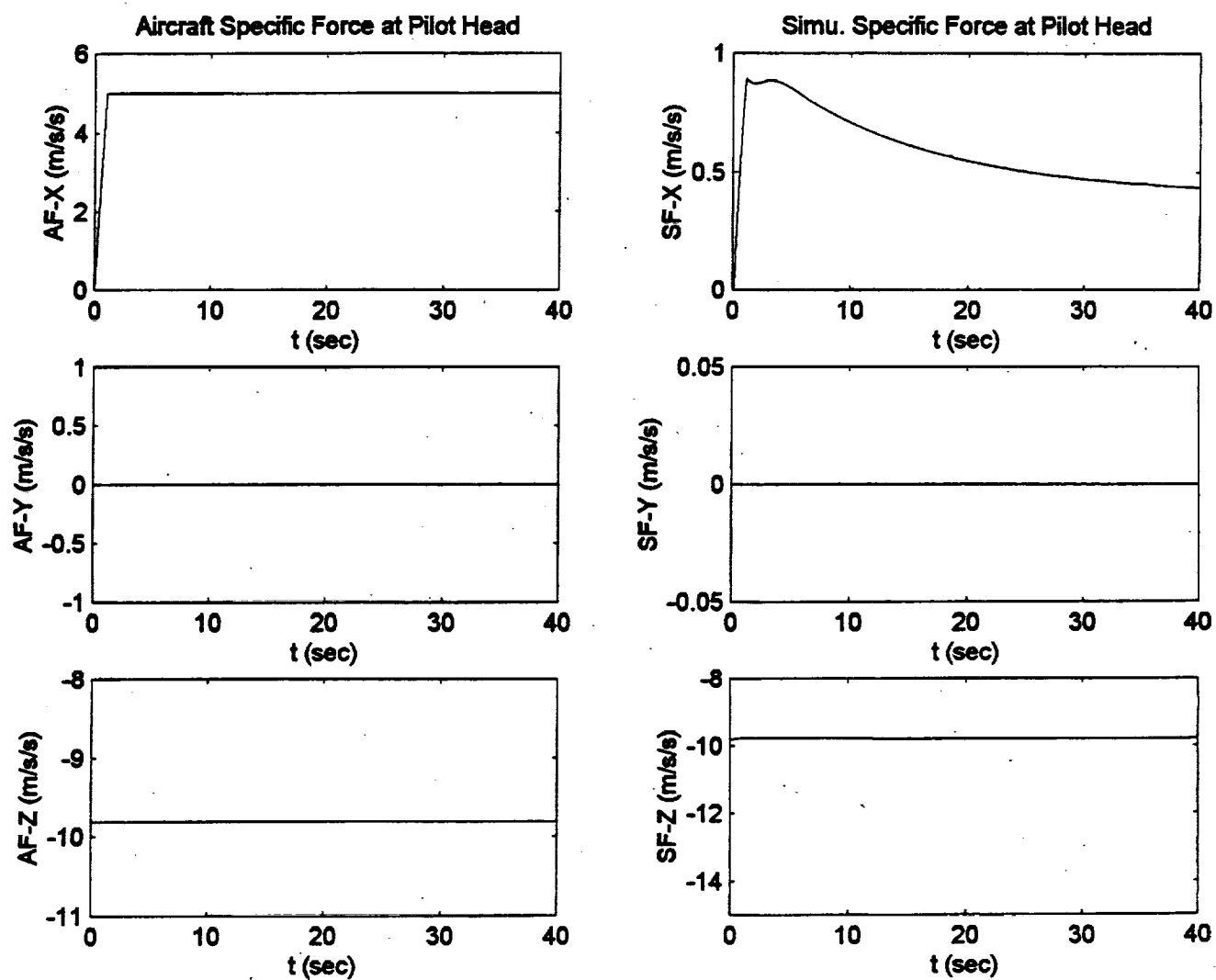


Figure C.7. Optimal Algorithm Surge 5 m/s²/s Ramp-Step Input.

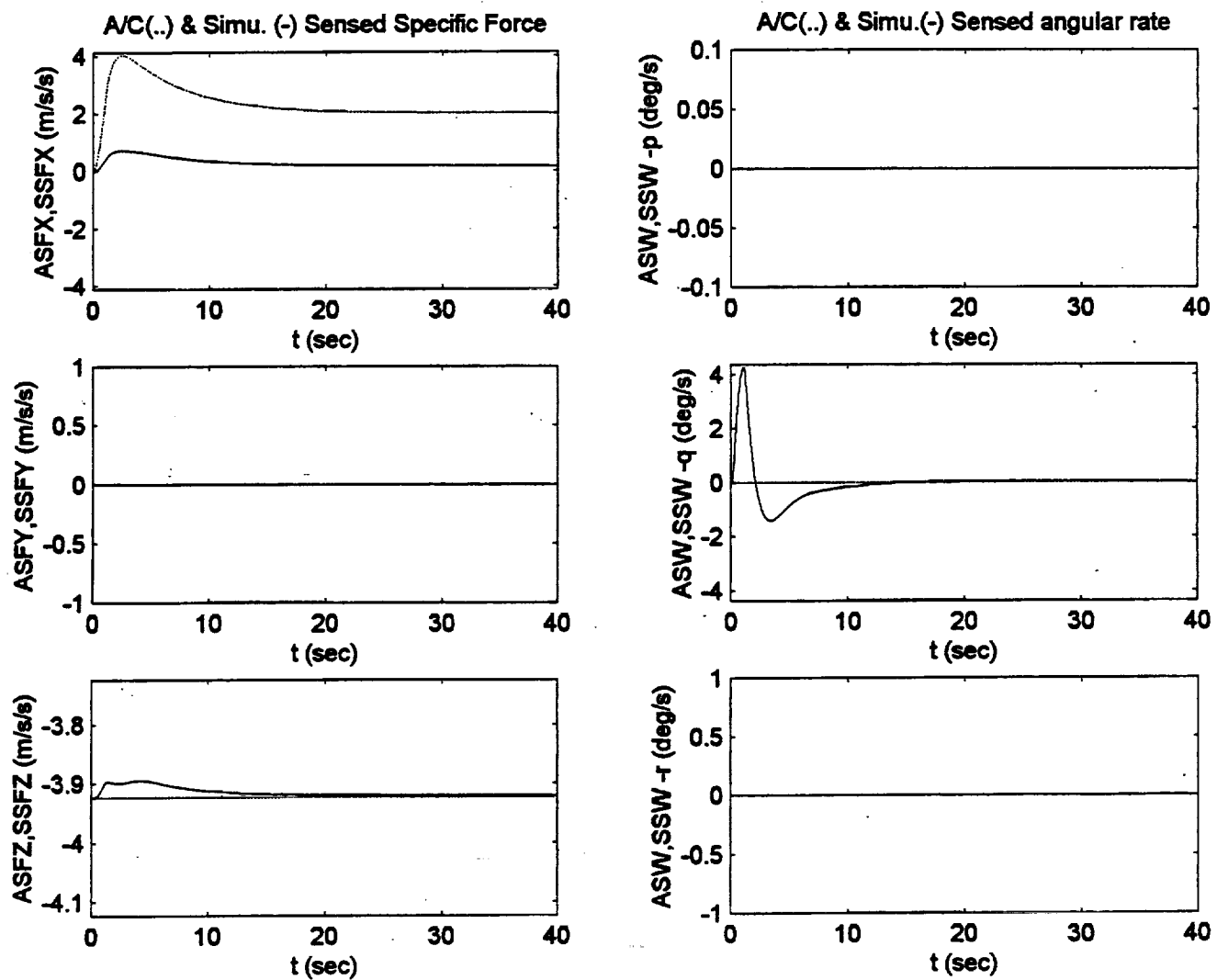


Figure C.7. Optimal Algorithm Surge $5 \text{ m/s}^2/\text{s}$ Ramp-Step Input.

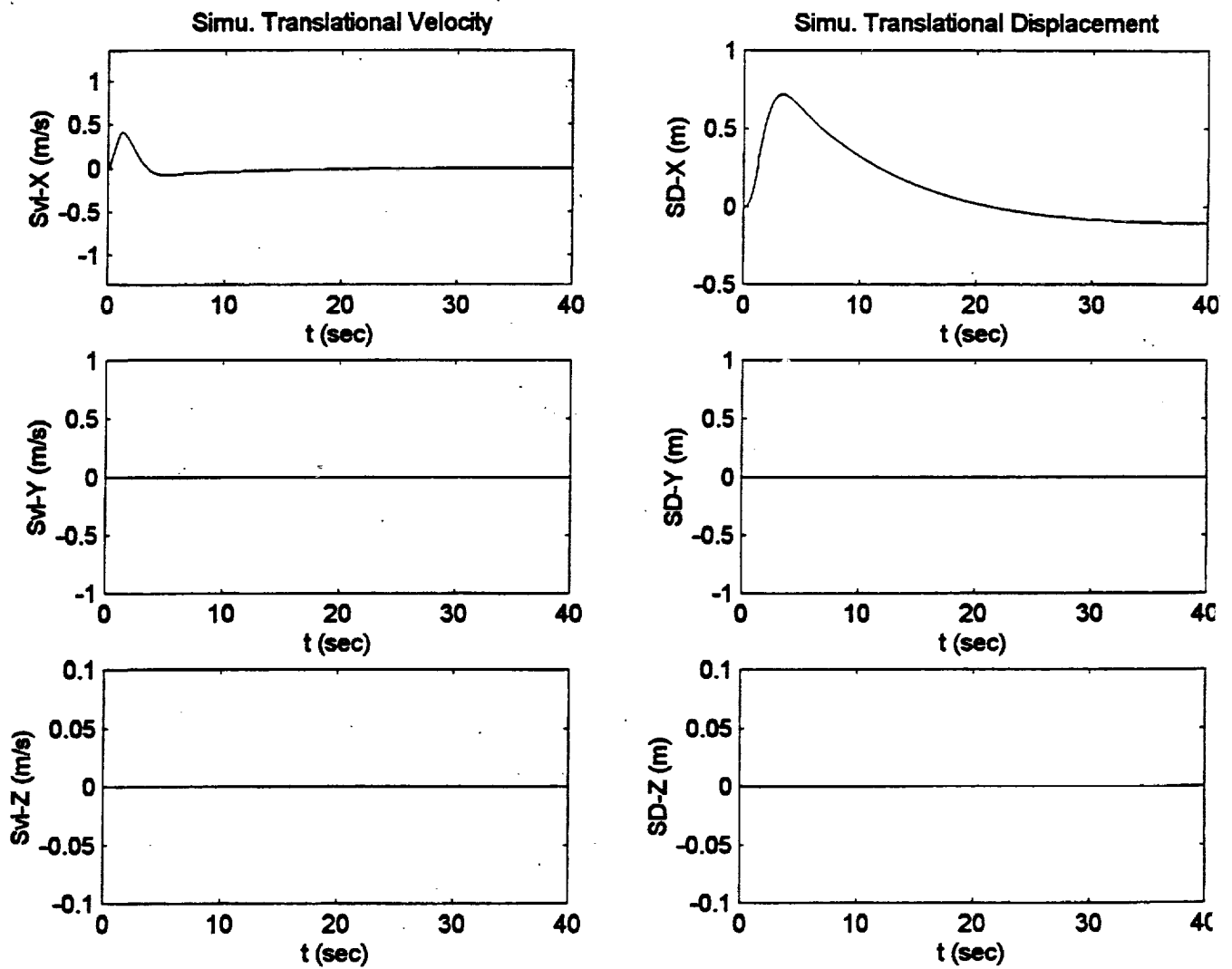


Figure C.7. Optimal Algorithm Surge $5 \text{ m/s}^2/\text{s}$ Ramp-Step Input.

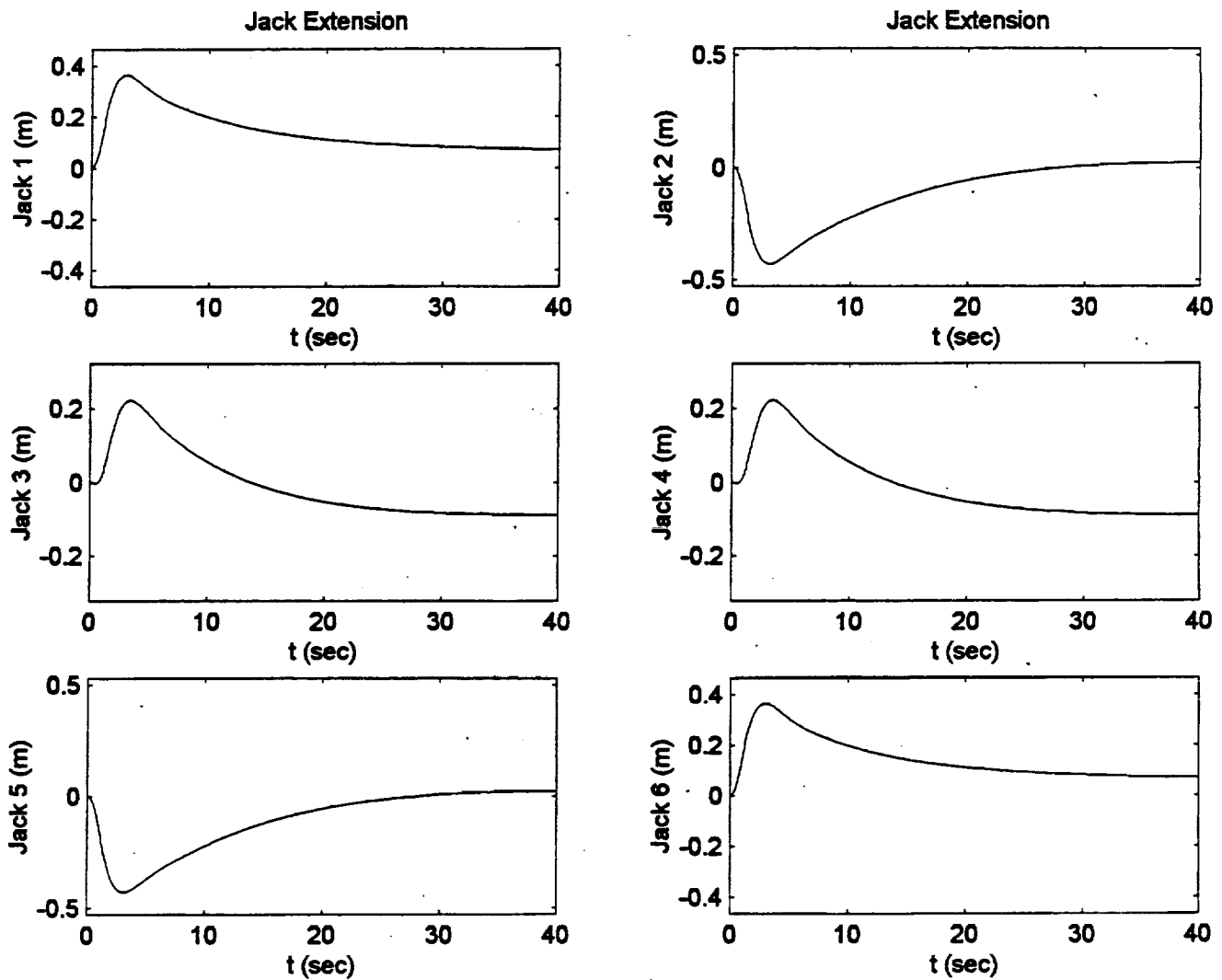


Figure C.7. Optimal Algorithm Surge $5 \text{ m/s}^2/\text{s}$ Ramp-Step Input.

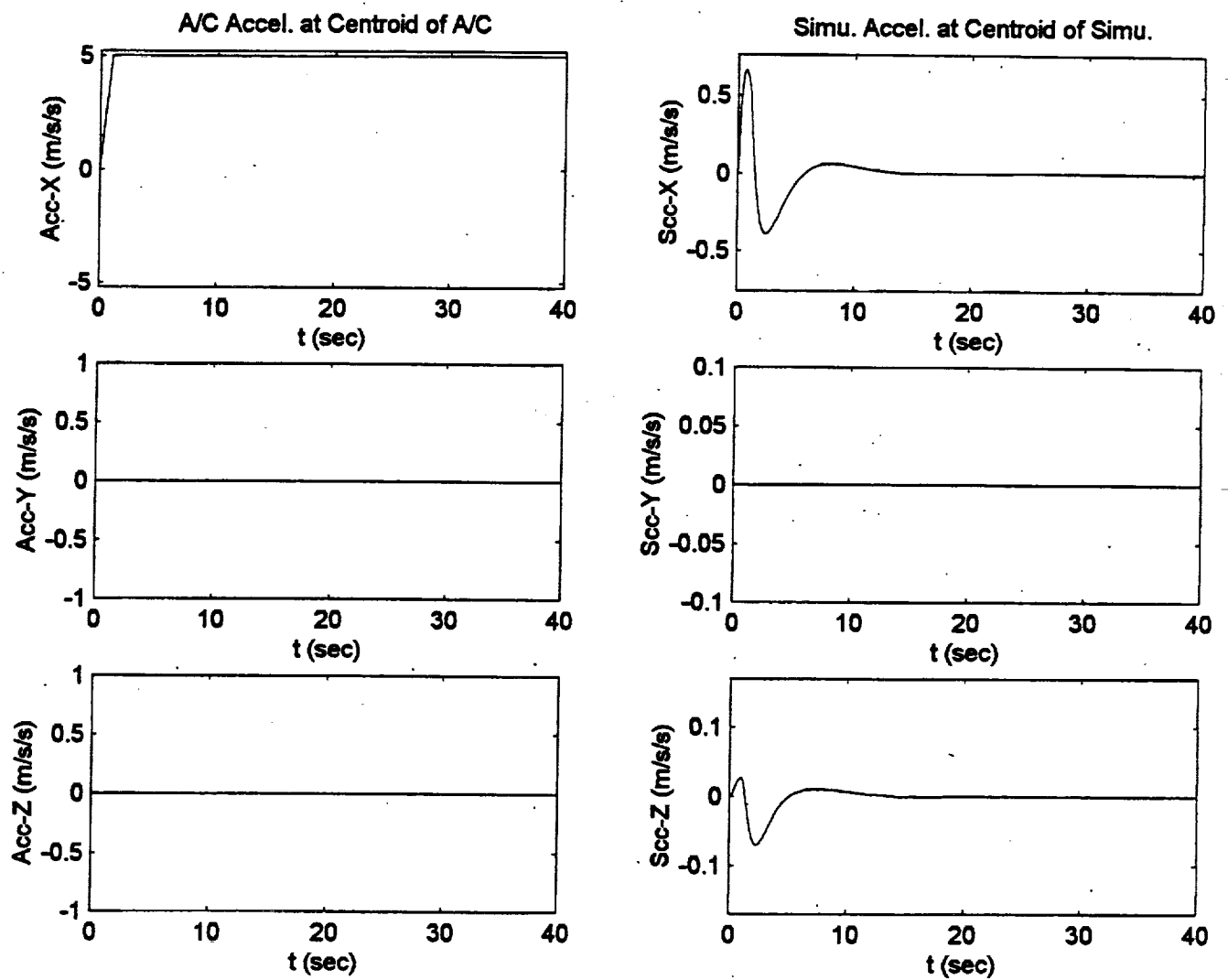


Figure C.8. NASA Adaptive Algorithm Surge 5 m/s²/s Ramp-Step Input.

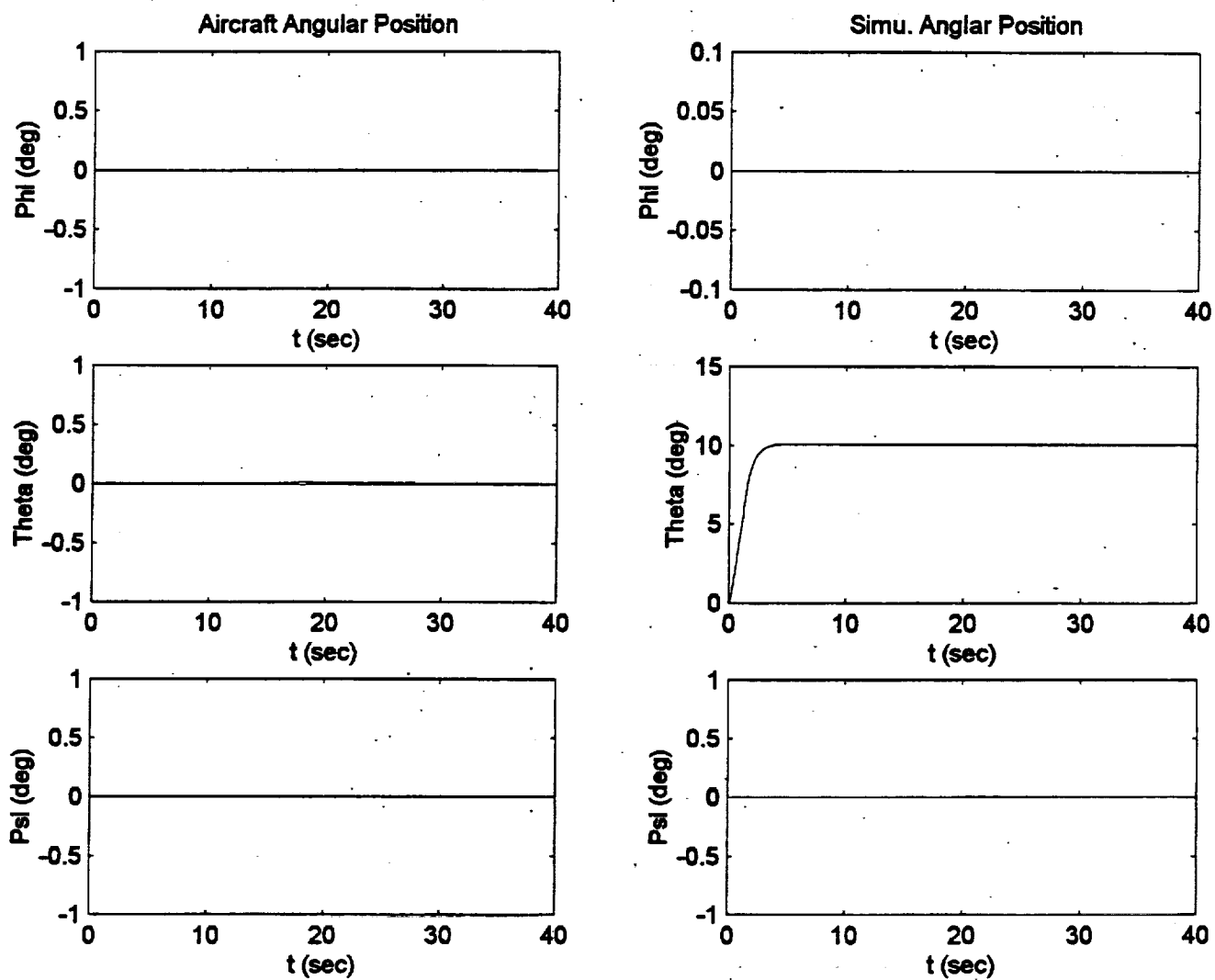


Figure C.8. NASA Adaptive Algorithm Surge $5 \text{ m/s}^2/\text{s}$ Ramp-Step Input.

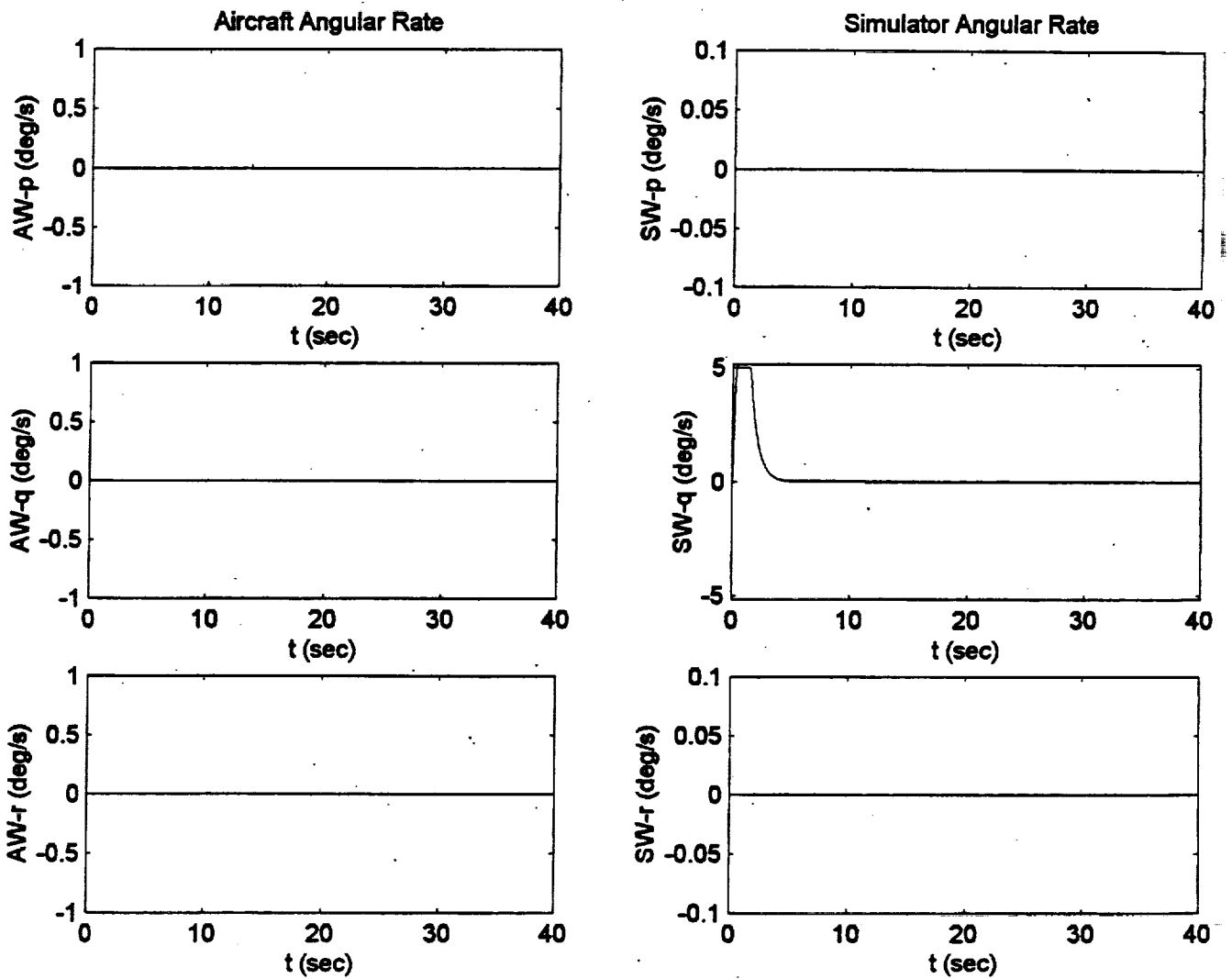


Figure C.8. NASA Adaptive Algorithm Surge $5 \text{ m/s}^2/\text{s}$ Ramp-Step Input.

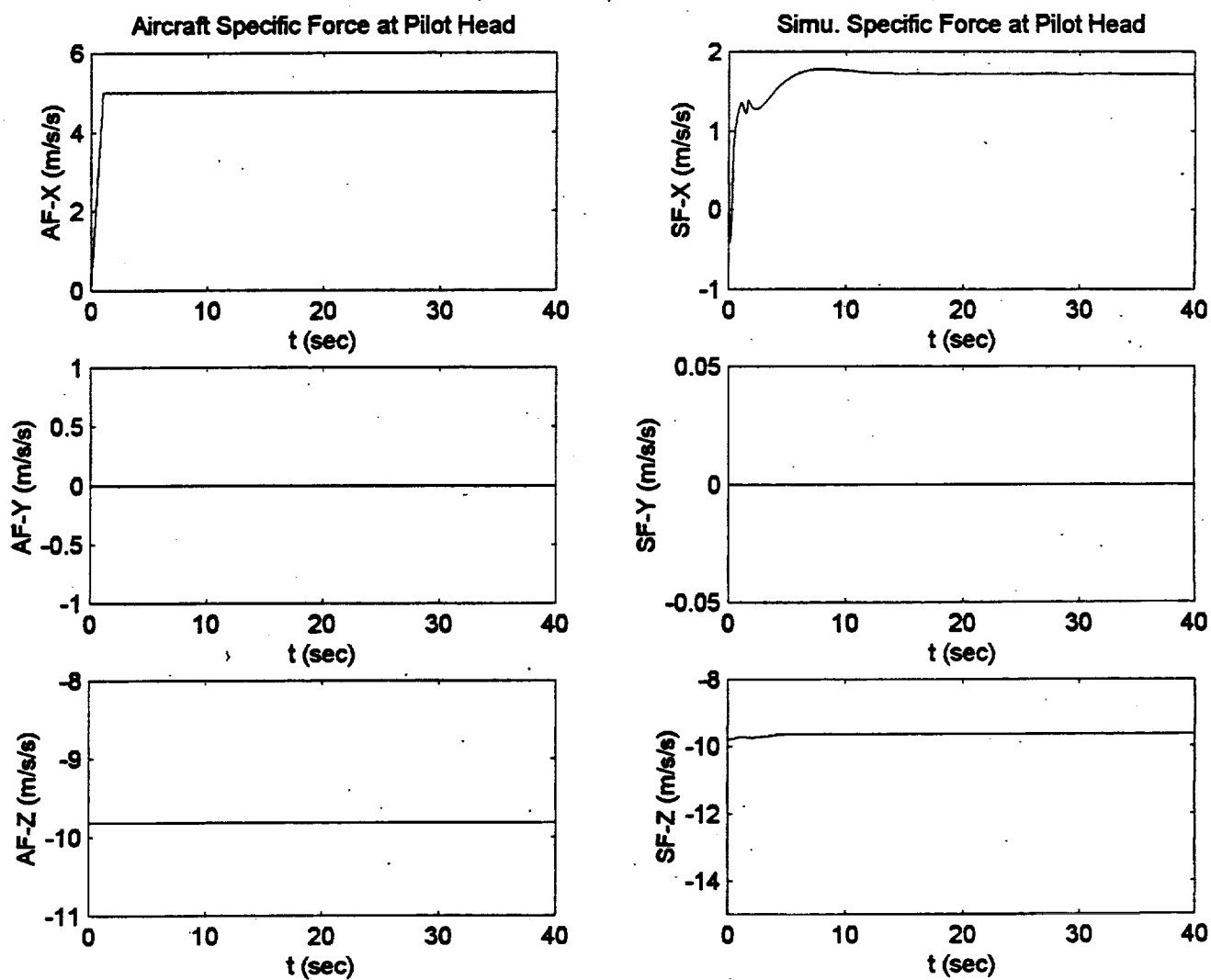


Figure C.8. NASA Adaptive Algorithm Surge 5 m/s²/s Ramp-Step Input.

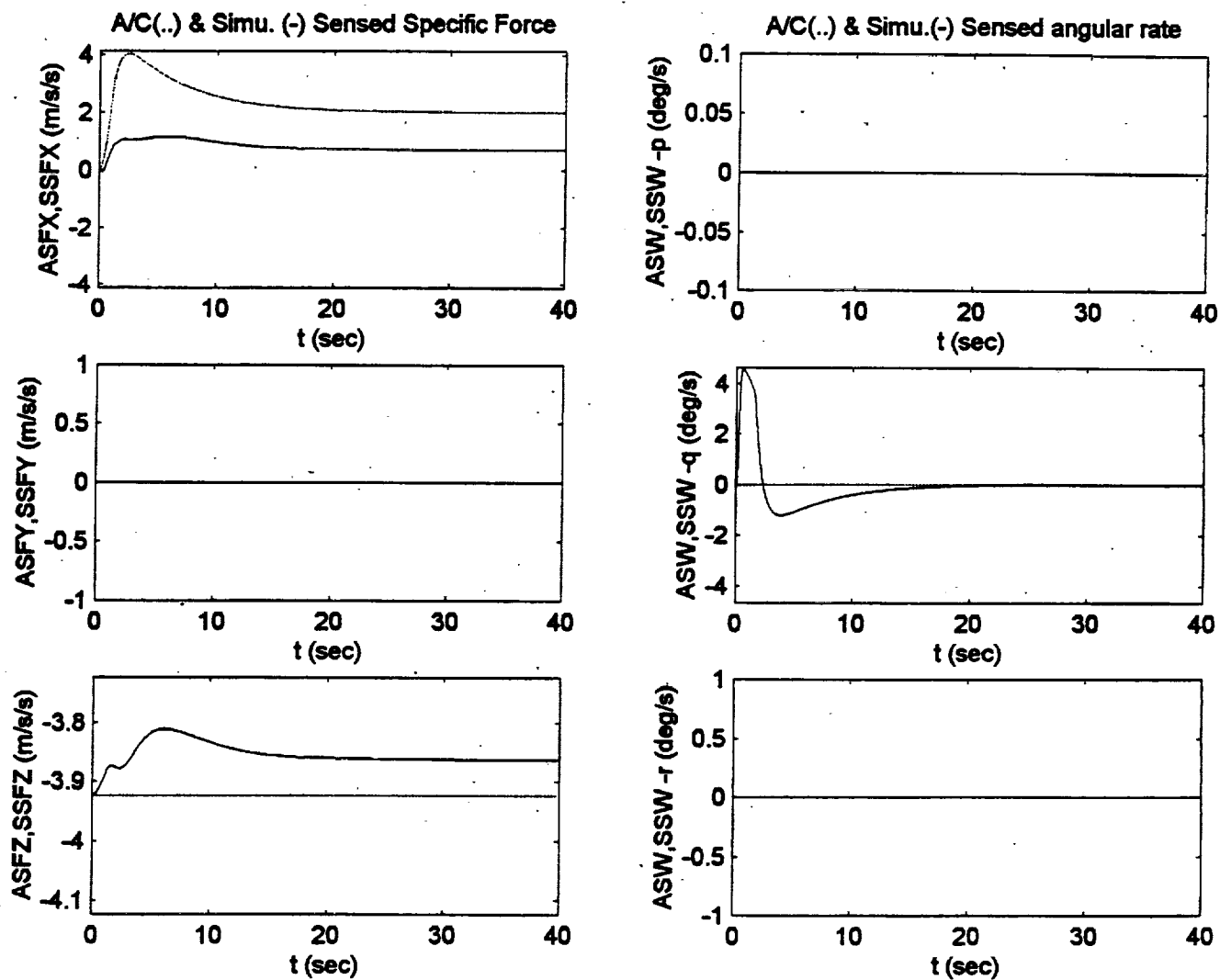


Figure C.8. NASA Adaptive Algorithm Surge 5 m/s²/s Ramp-Step Input.

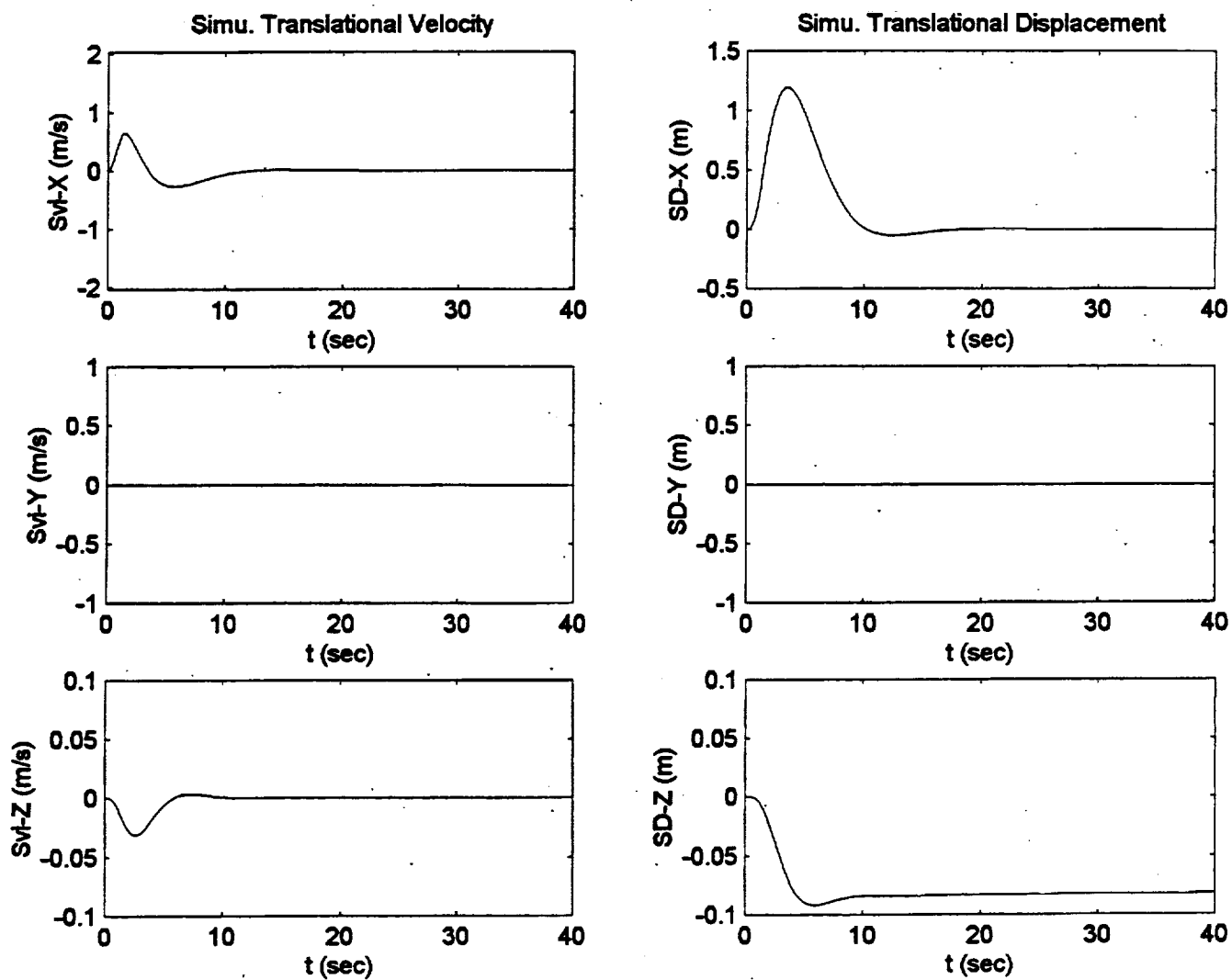


Figure C.8. NASA Adaptive Algorithm Surge $5 \text{ m/s}^2/\text{s}$ Ramp-Step Input.

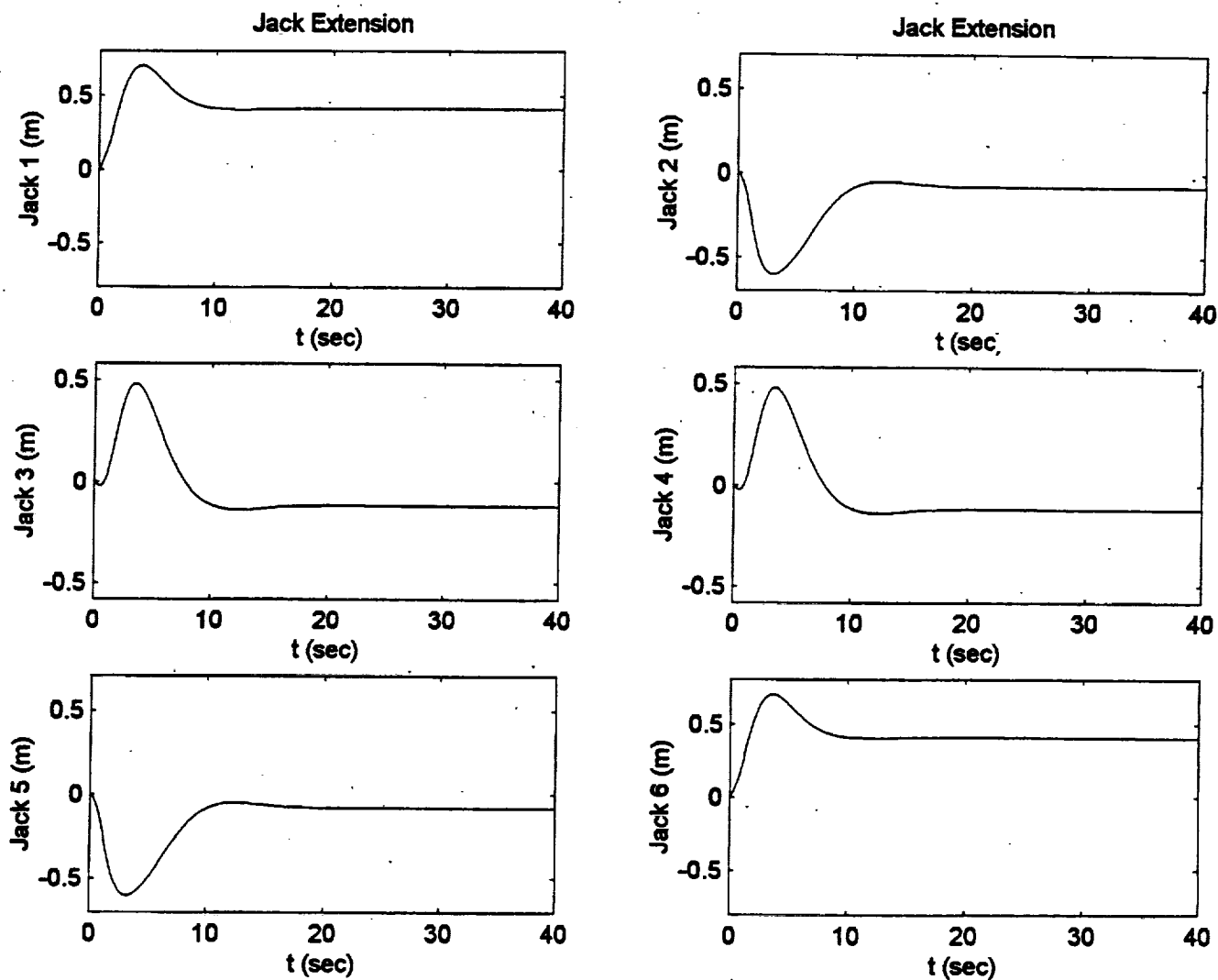


Figure C.8. NASA Adaptive Algorithm Surge $5 \text{ m/s}^2/\text{s}$ Ramp-Step Input.

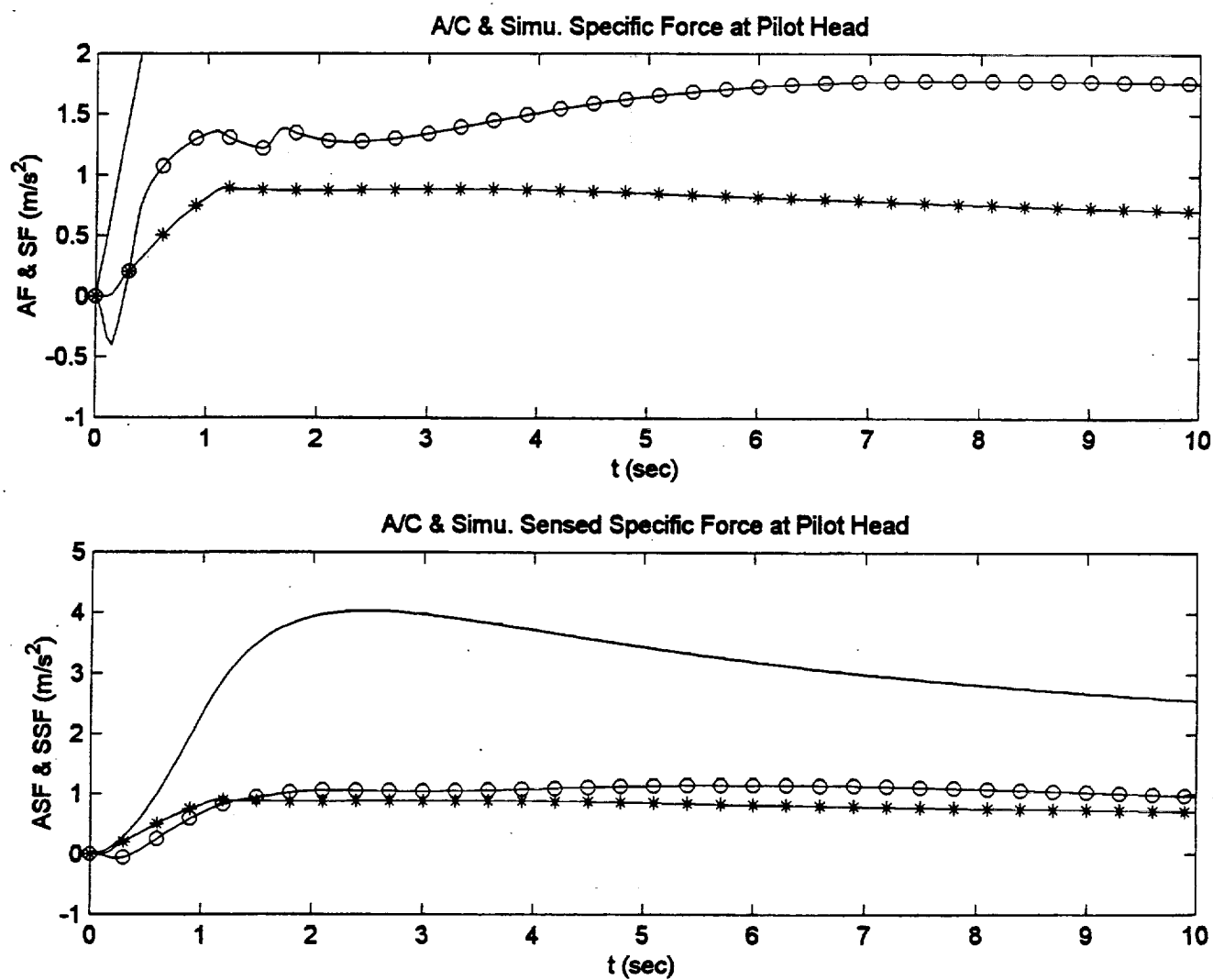


Figure C.9. Surge $5 \text{ m/s}^2/\text{s}$ Ramp-Step Input Specific Force Comparison.

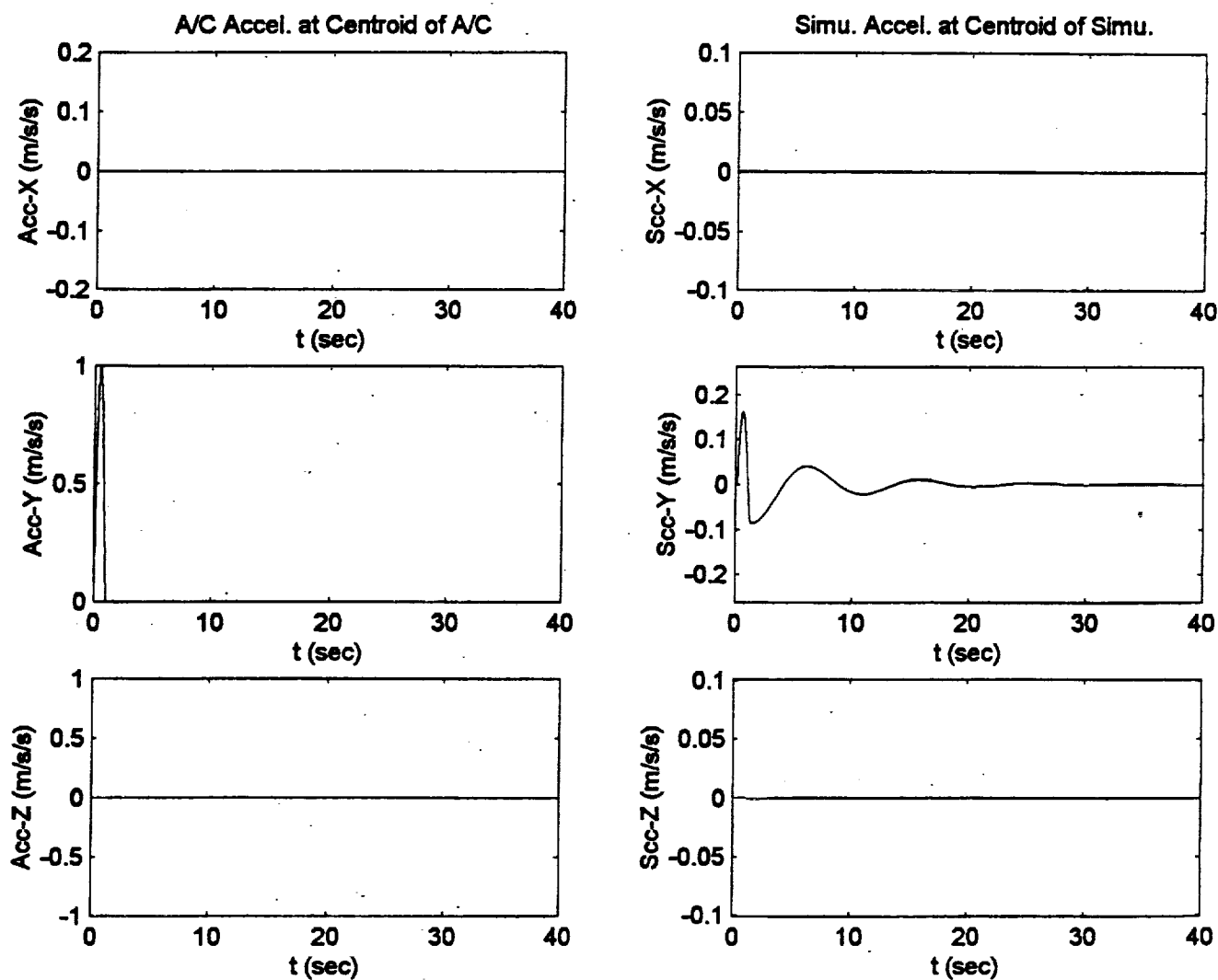


Figure C.10. Optimal Algorithm Sway Half Sinusoidal Pulse Input.

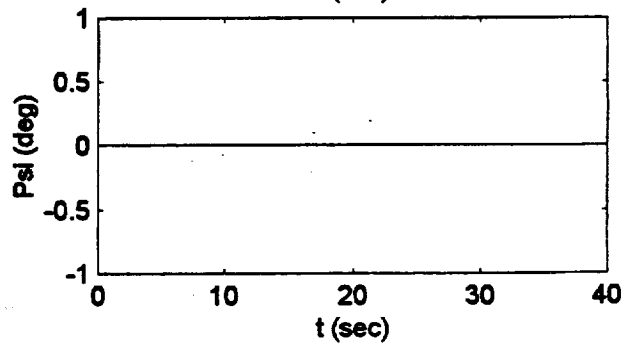
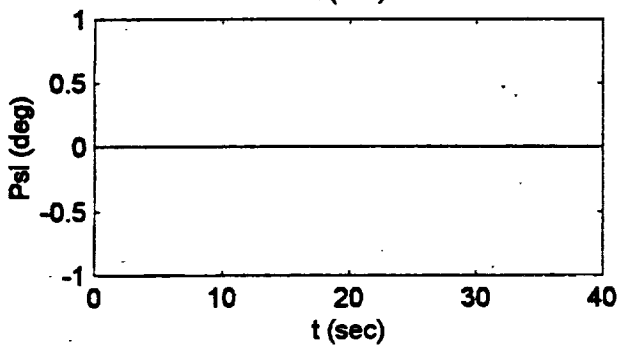
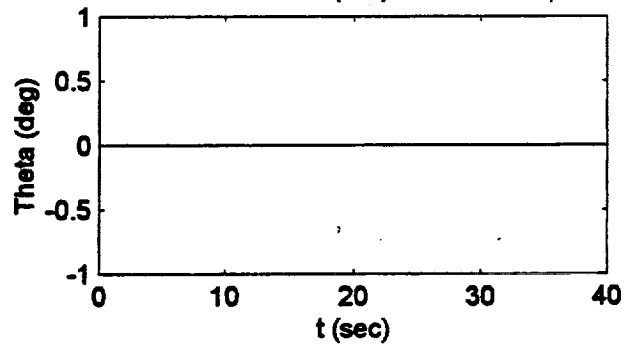
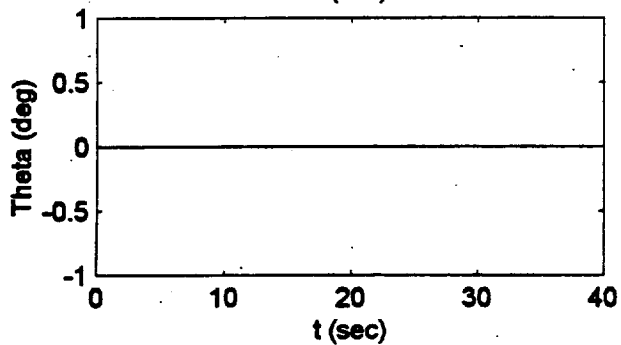
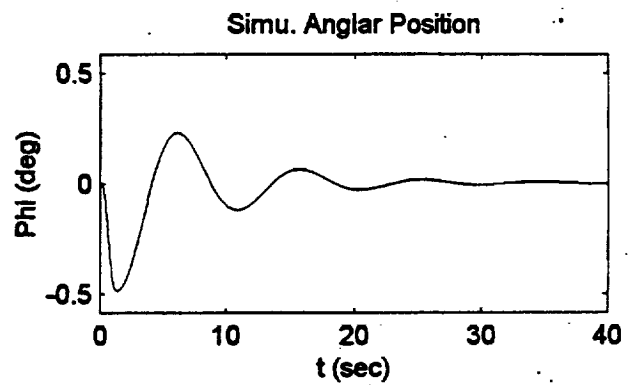
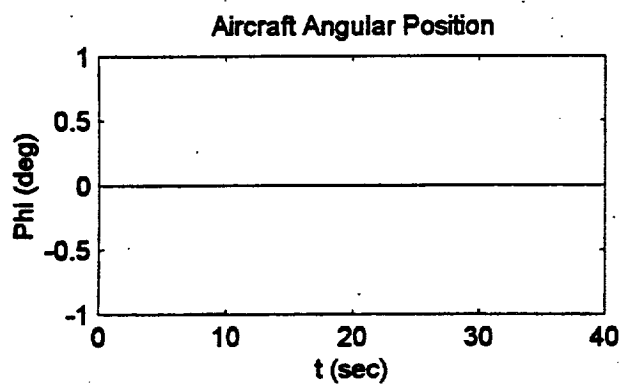


Figure C.10. Optimal Algorithm Sway Half Sinusoidal Pulse Input.

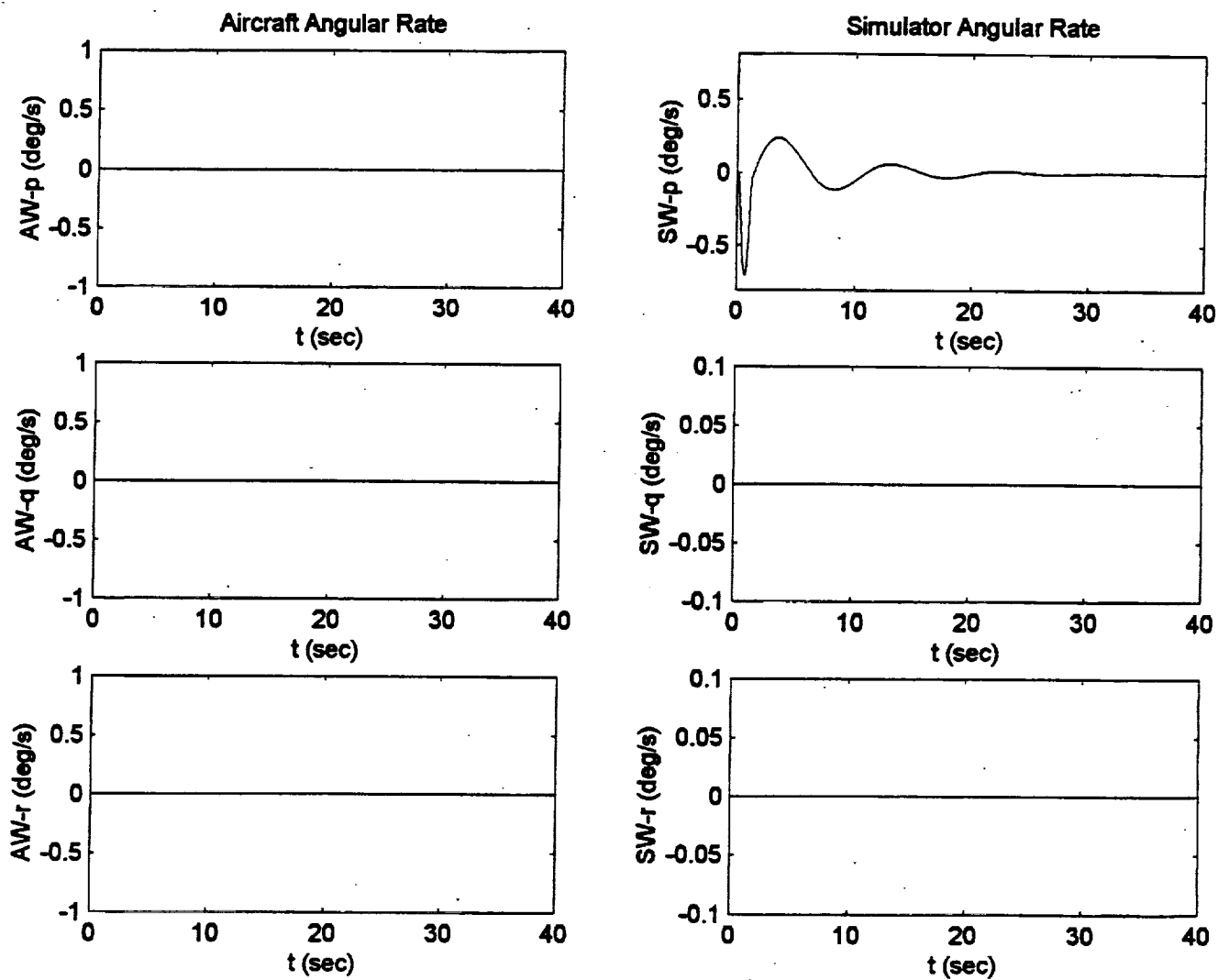


Figure C.10. Optimal Algorithm Sway Half Sinusoidal Pulse Input.

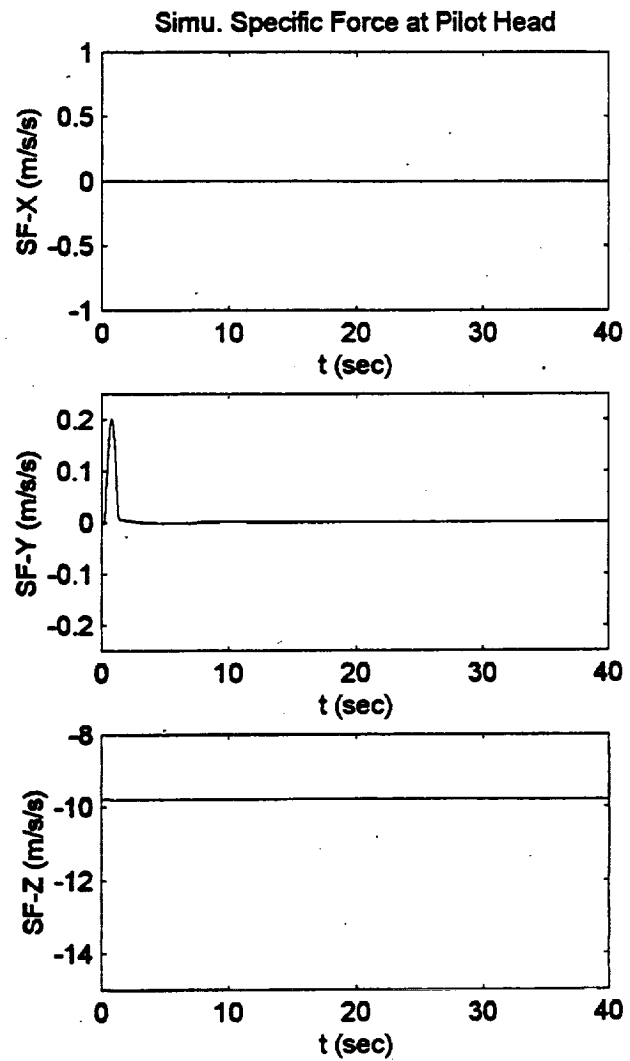
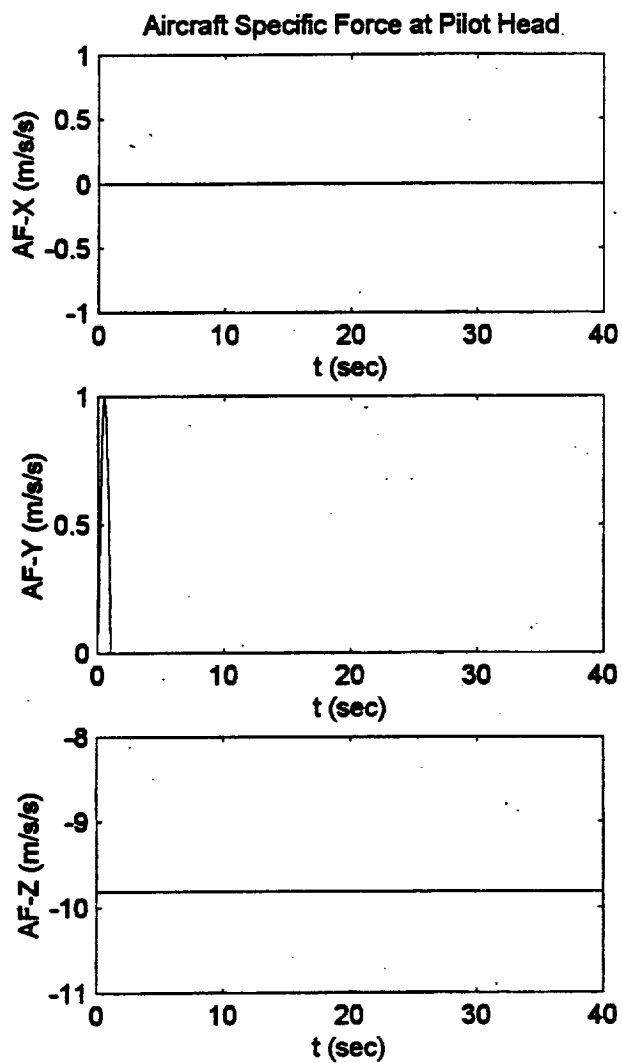


Figure C.10. Optimal Algorithm Sway Half Sinusoidal Pulse Input.

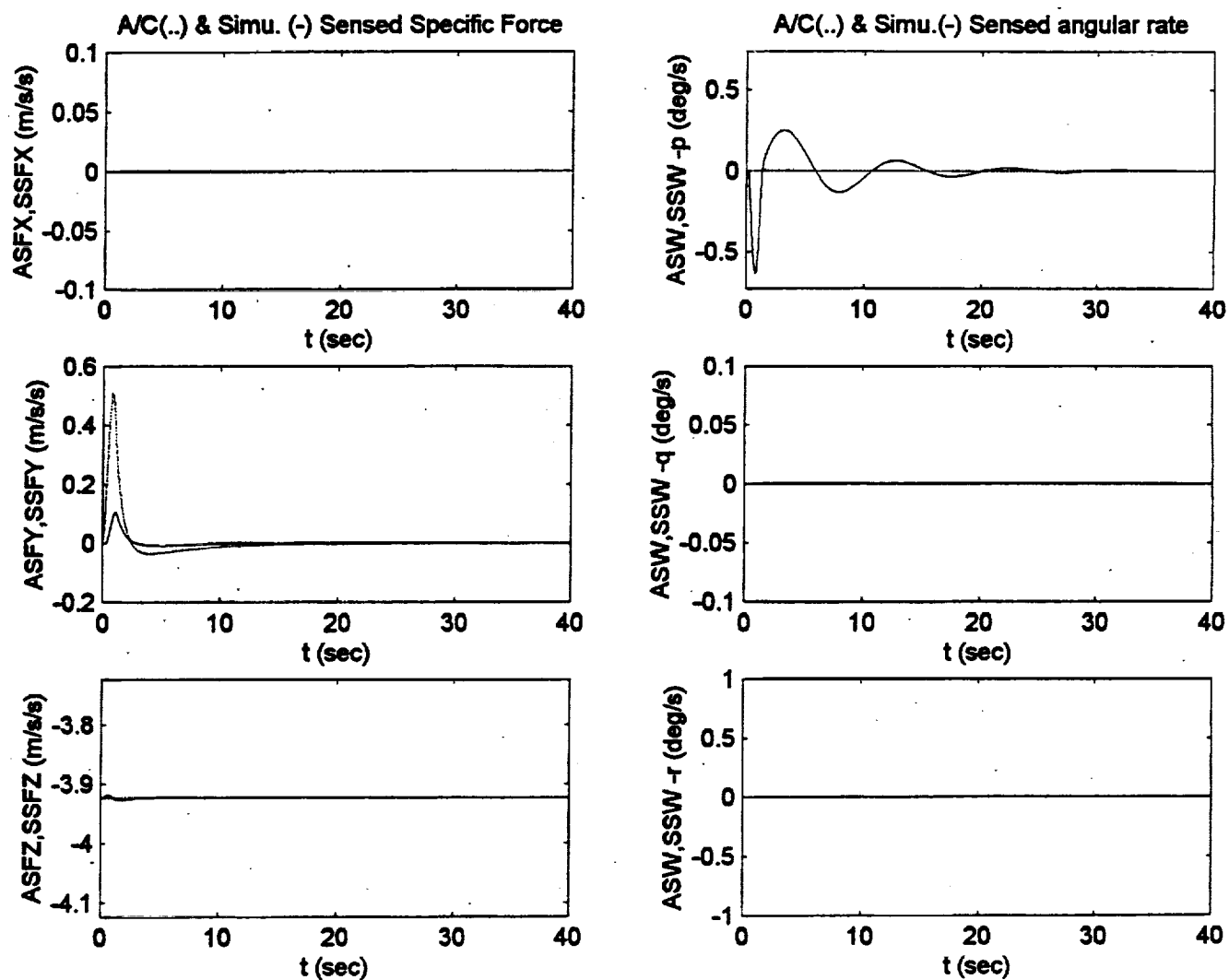


Figure C.10. Optimal Algorithm Sway Half Sinusoidal Pulse Input.

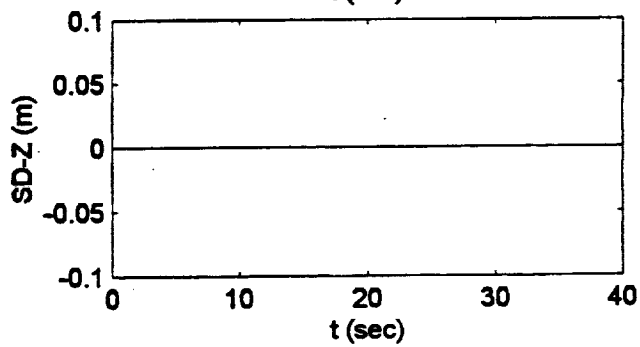
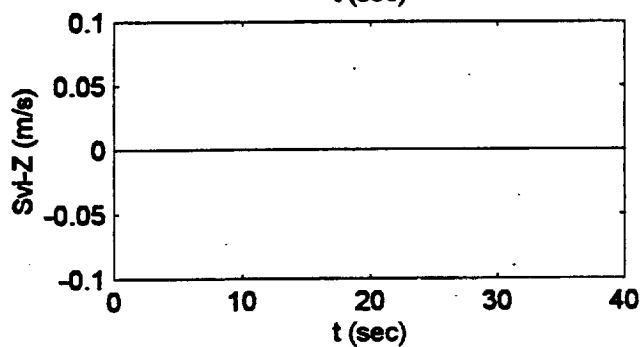
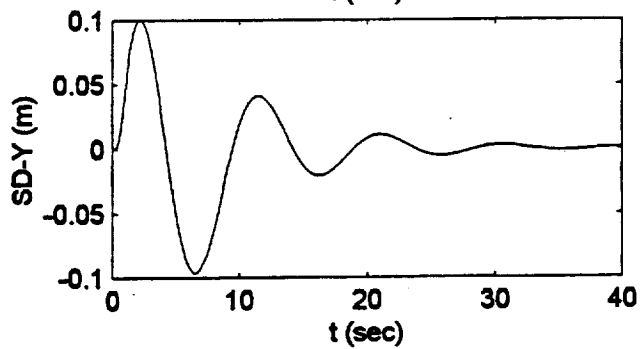
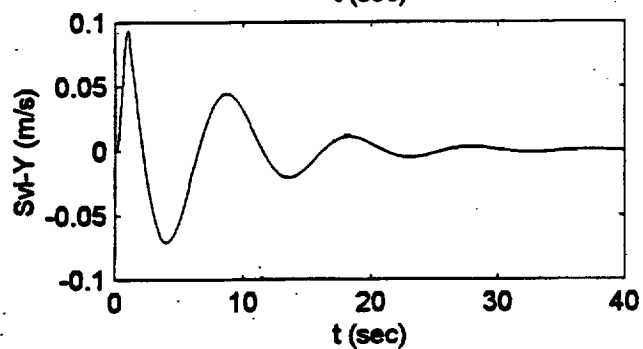
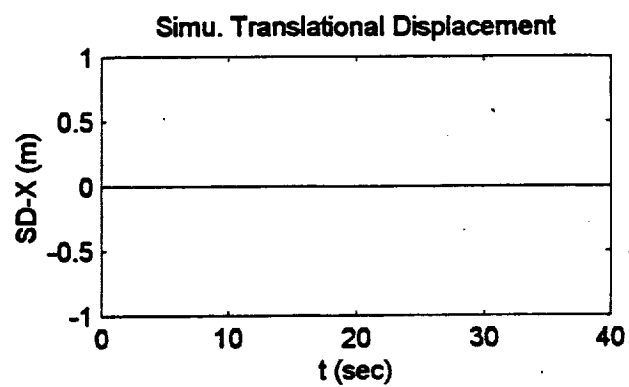
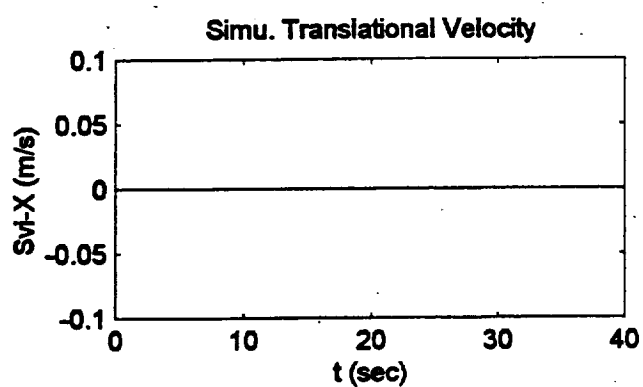


Figure C.10. Optimal Algorithm Sway Half Sinusoidal Pulse Input.

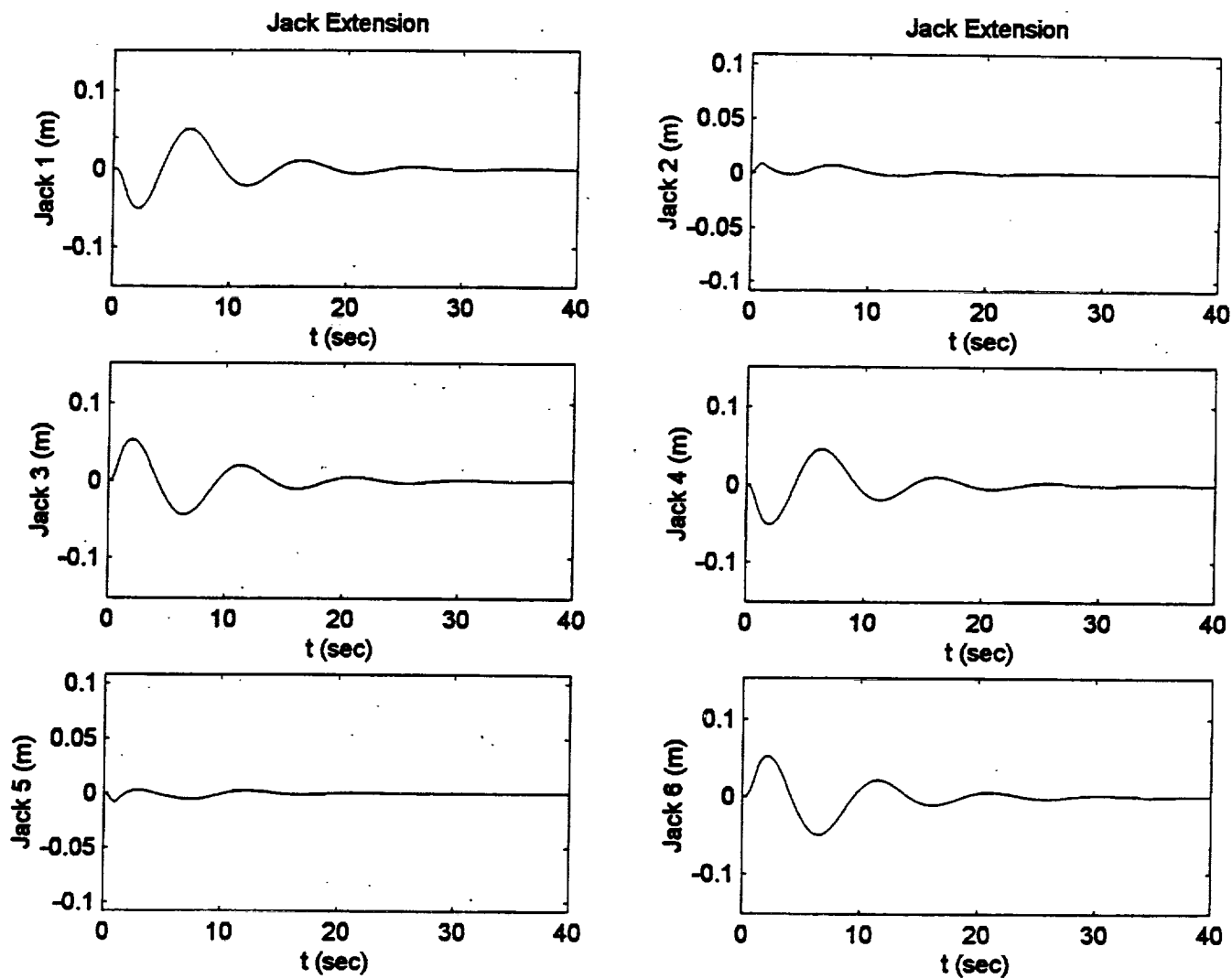


Figure C.10. Optimal Algorithm Sway Half Sinusoidal Pulse Input.

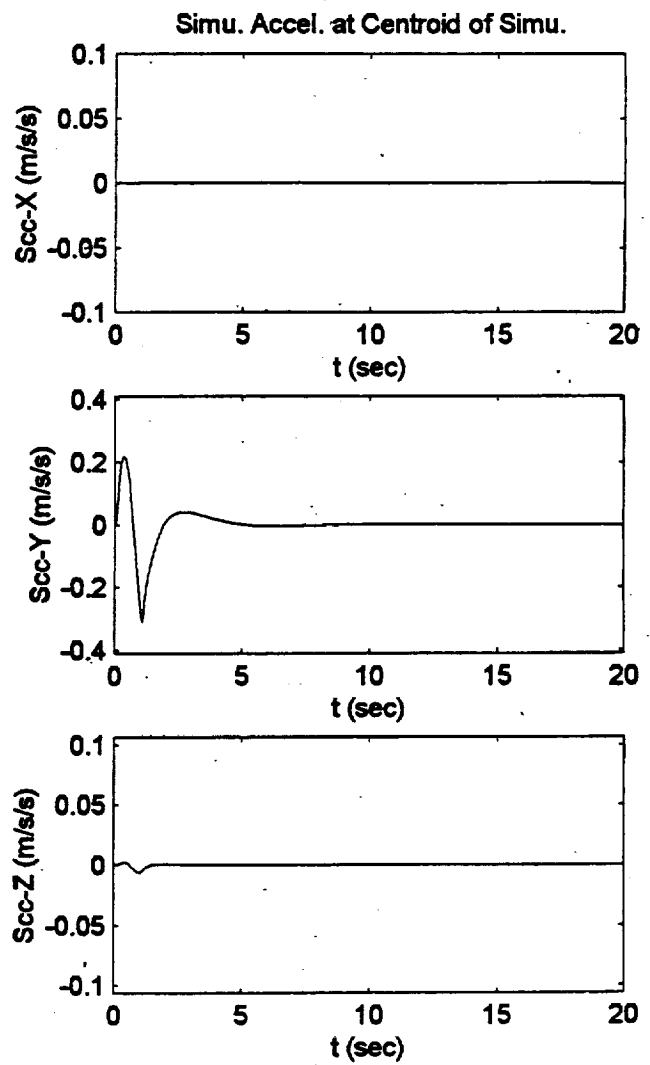
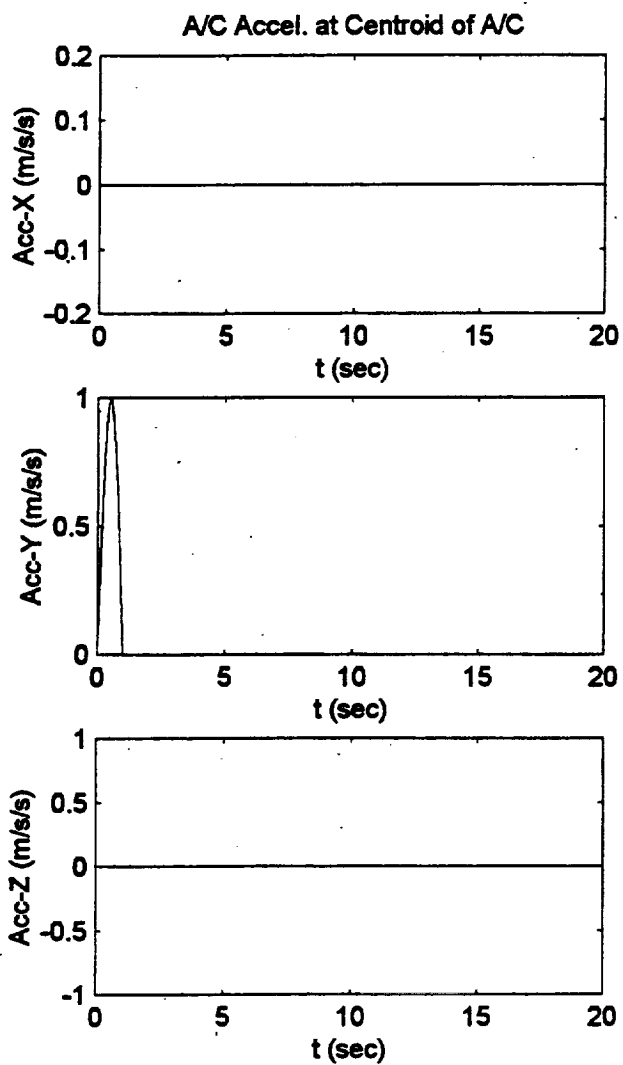


Figure C.11. NASA Adaptive Algorithm Sway Half Sinusoidal Pulse Input.

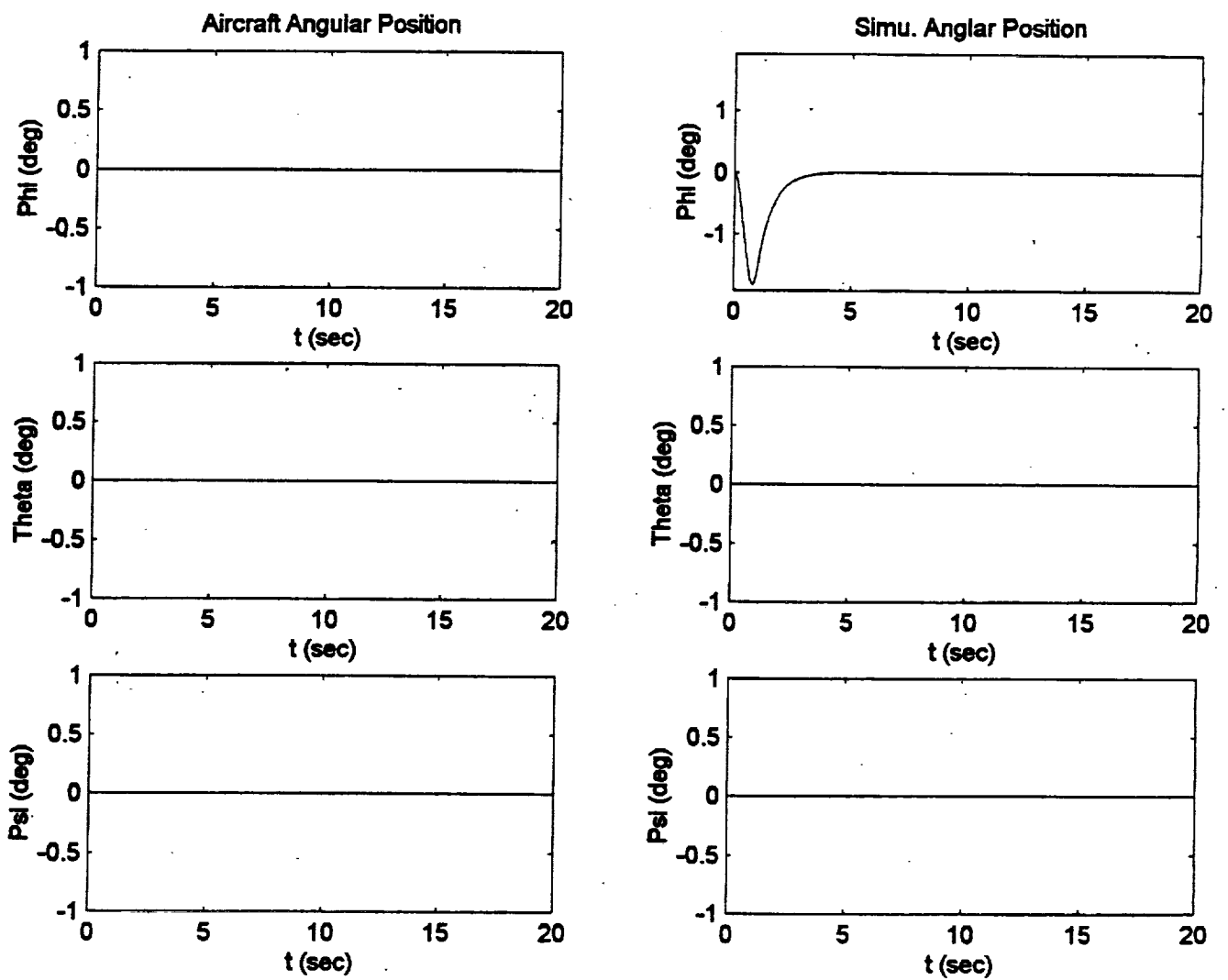


Figure C.11. NASA Adaptive Algorithm Sway Half Sinusoidal Pulse Input.

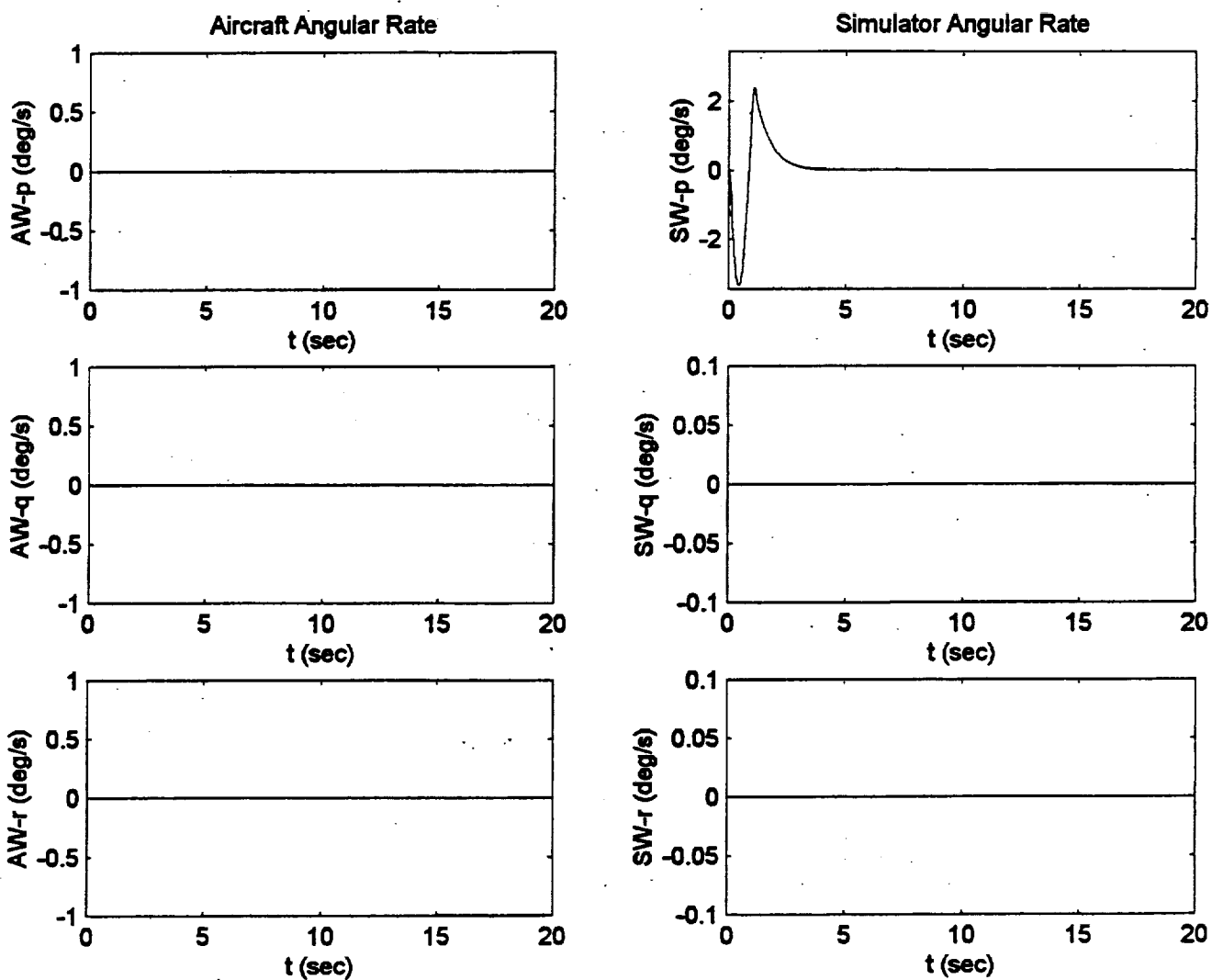


Figure C.11. NASA Adaptive Algorithm Sway Half Sinusoidal Pulse Input.

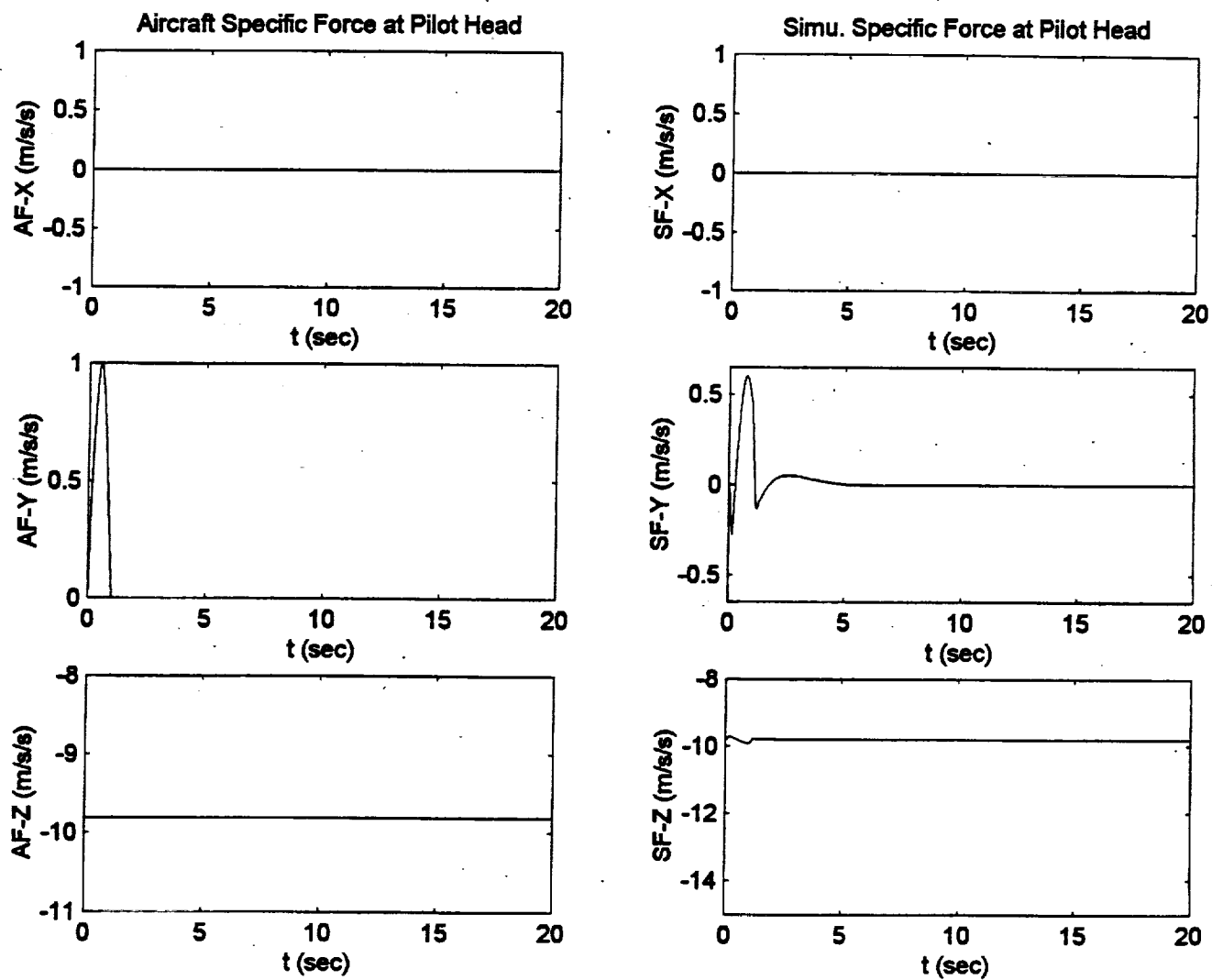


Figure C.11. NASA Adaptive Algorithm Sway Half Sinusoidal Pulse Input.

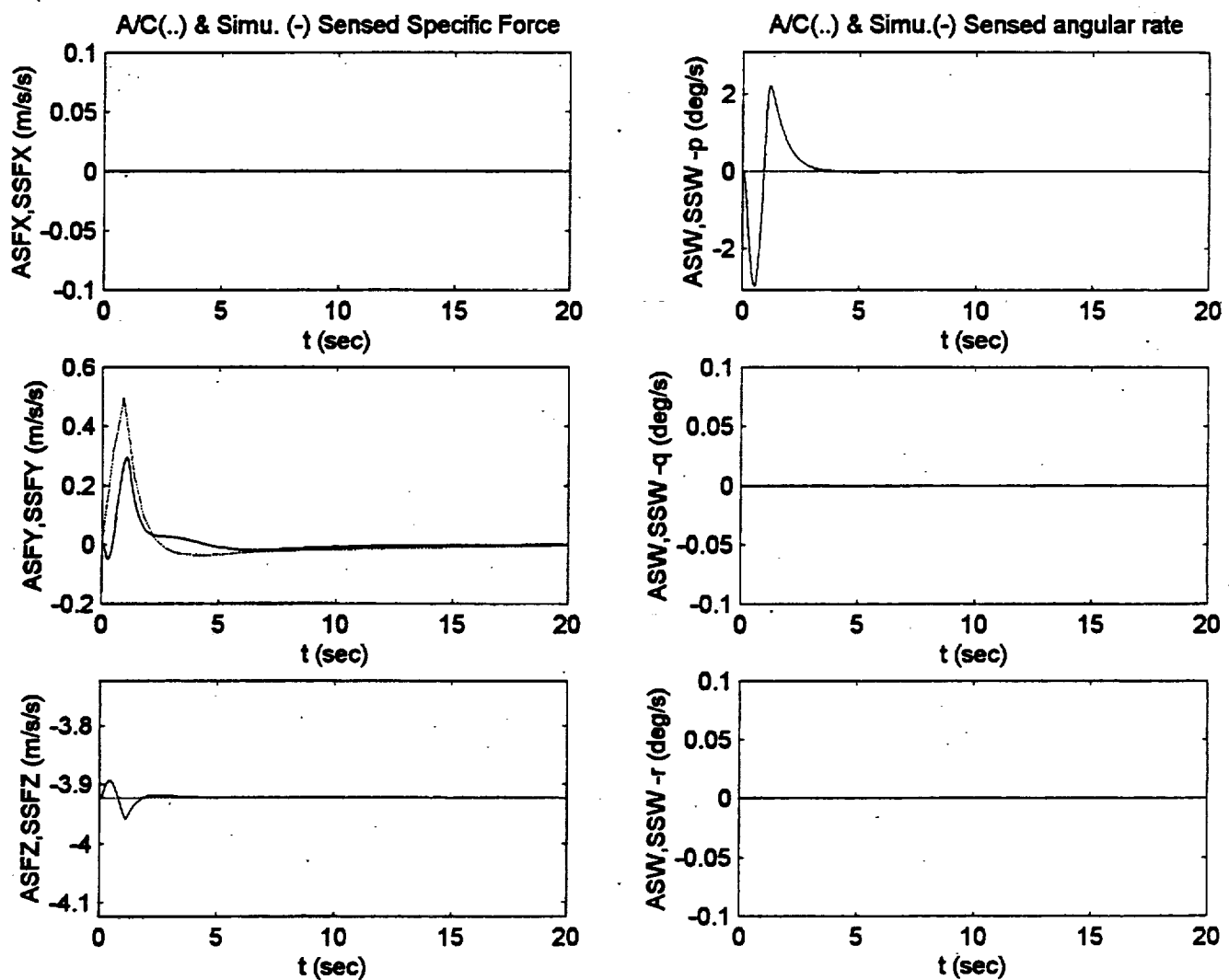


Figure C.11. NASA Adaptive Algorithm Sway Half Sinusoidal Pulse Input.

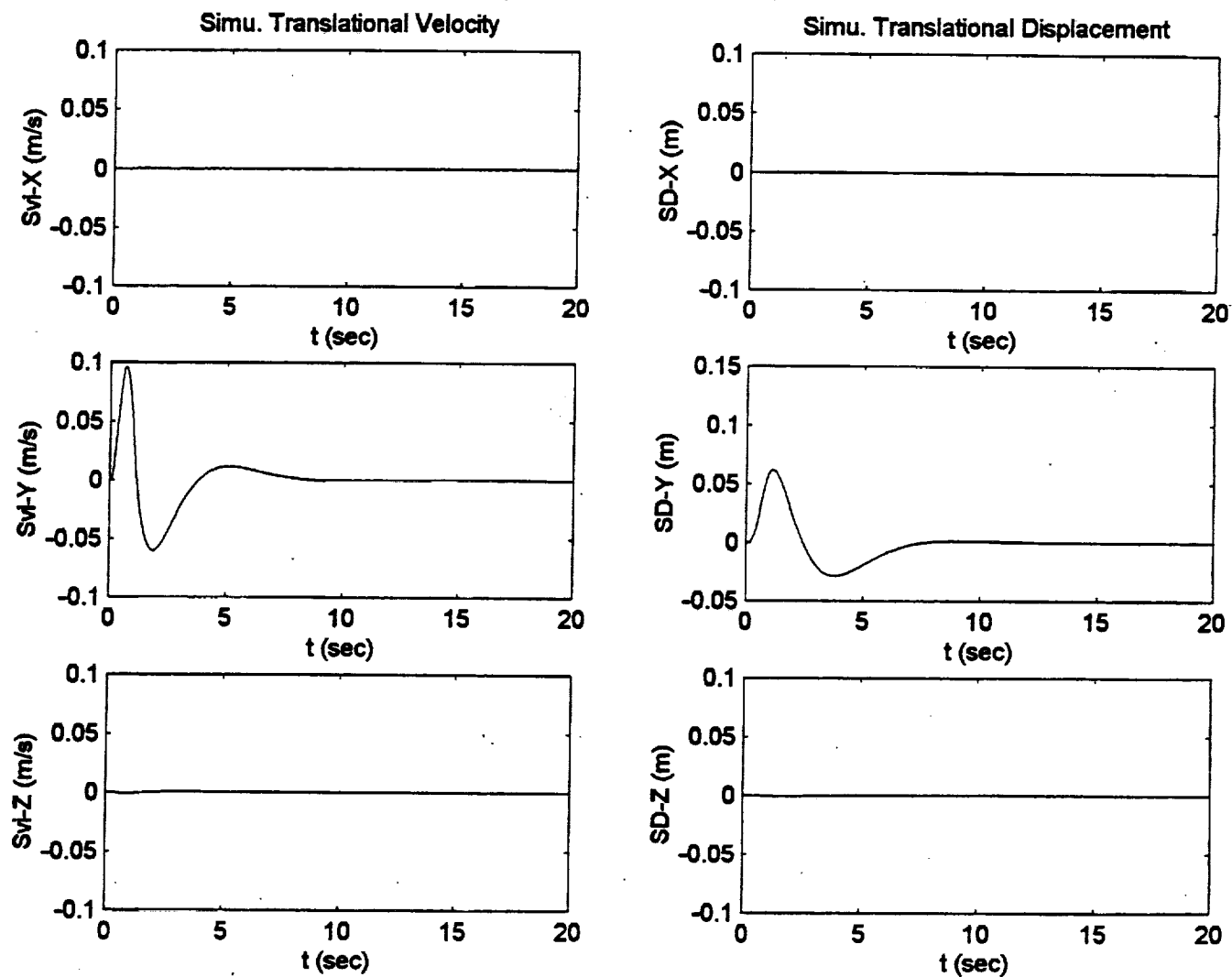


Figure C.11. NASA Adaptive Algorithm Sway Half Sinusoidal Pulse Input.

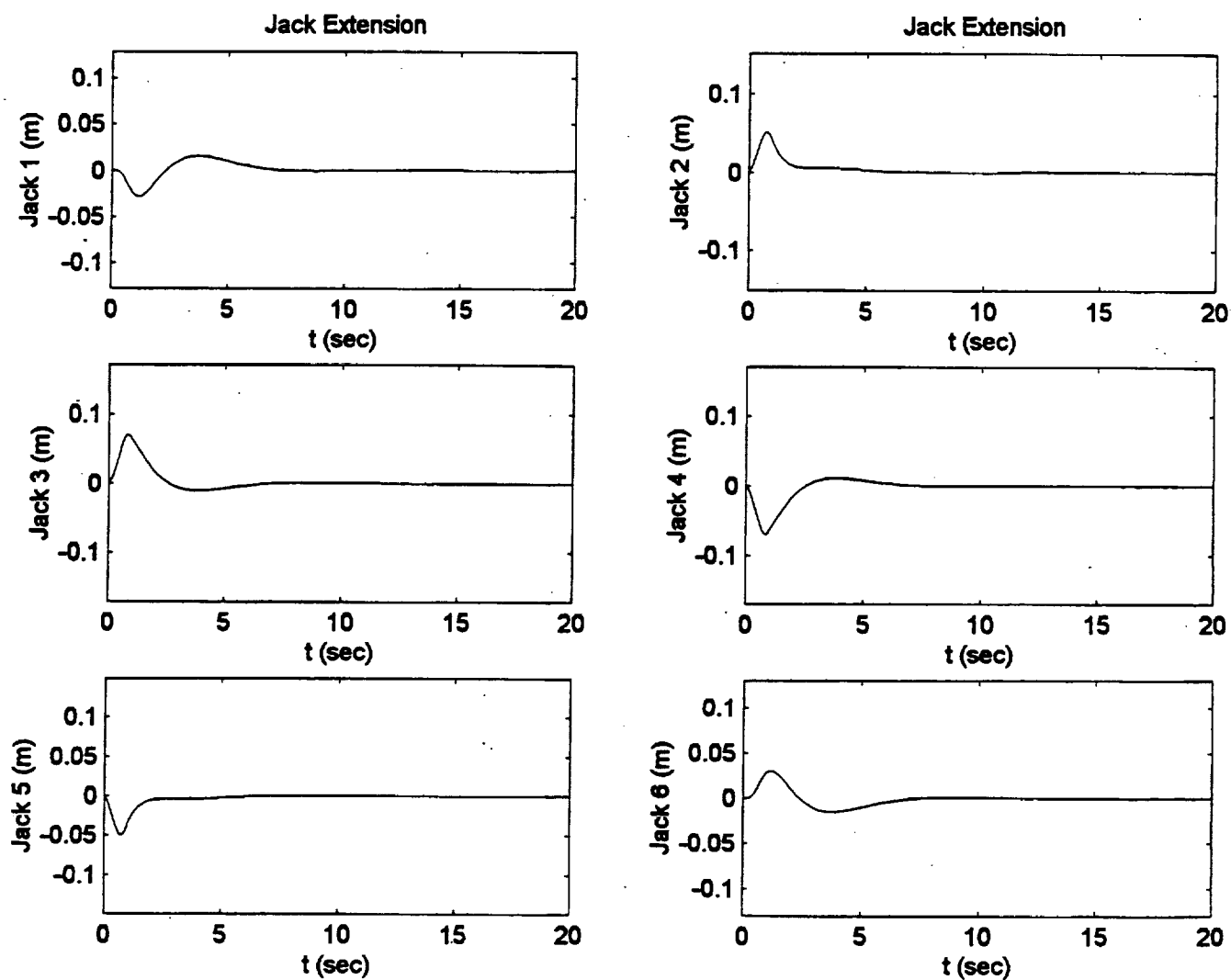


Figure C.11. NASA Adaptive Algorithm Sway Half Sinusoidal Pulse Input.

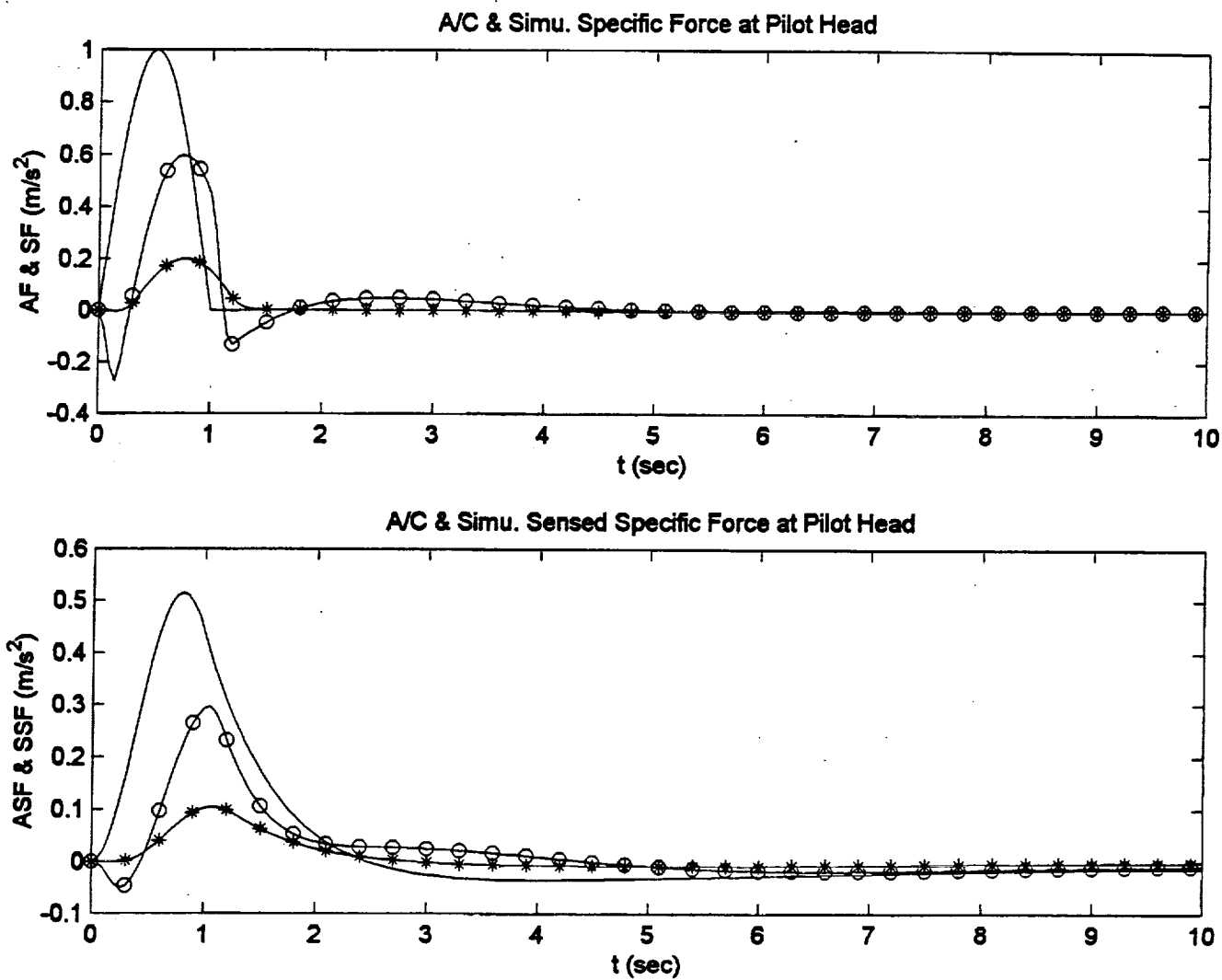


Figure C.12. Sway Half Sinusoidal Pulse Input Specific Force Comparison.

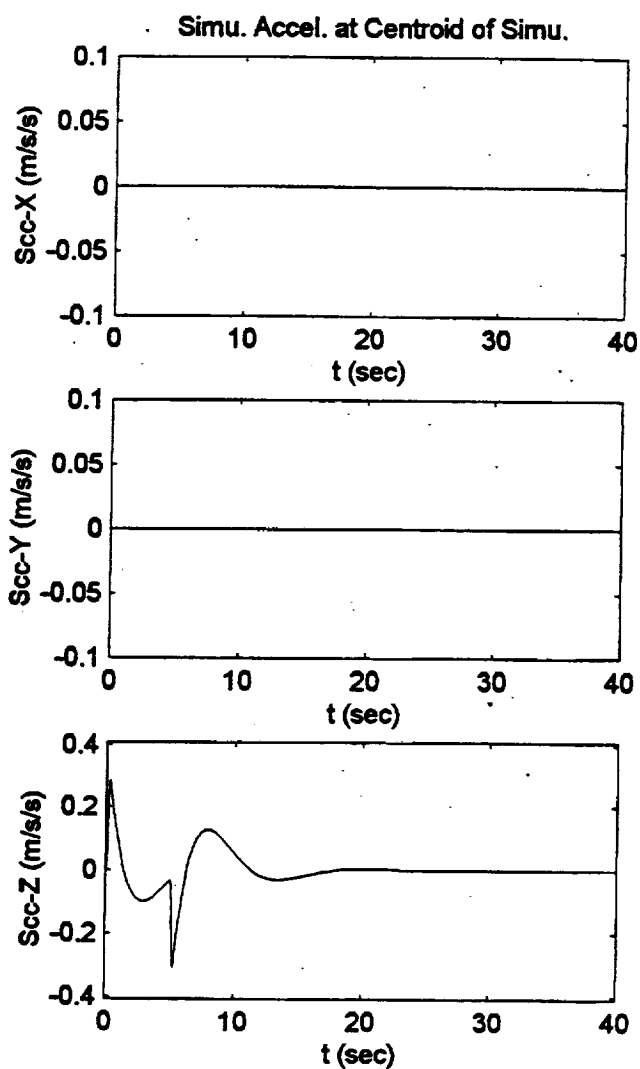
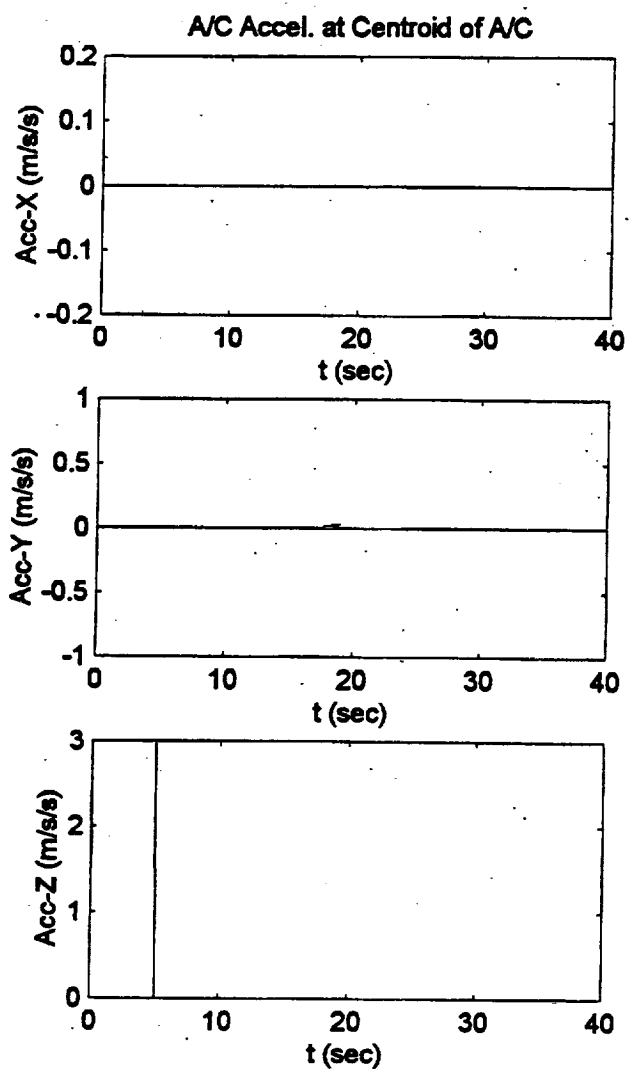


Figure C.13. Optimal Algorithm Heave Square Pulse Input.

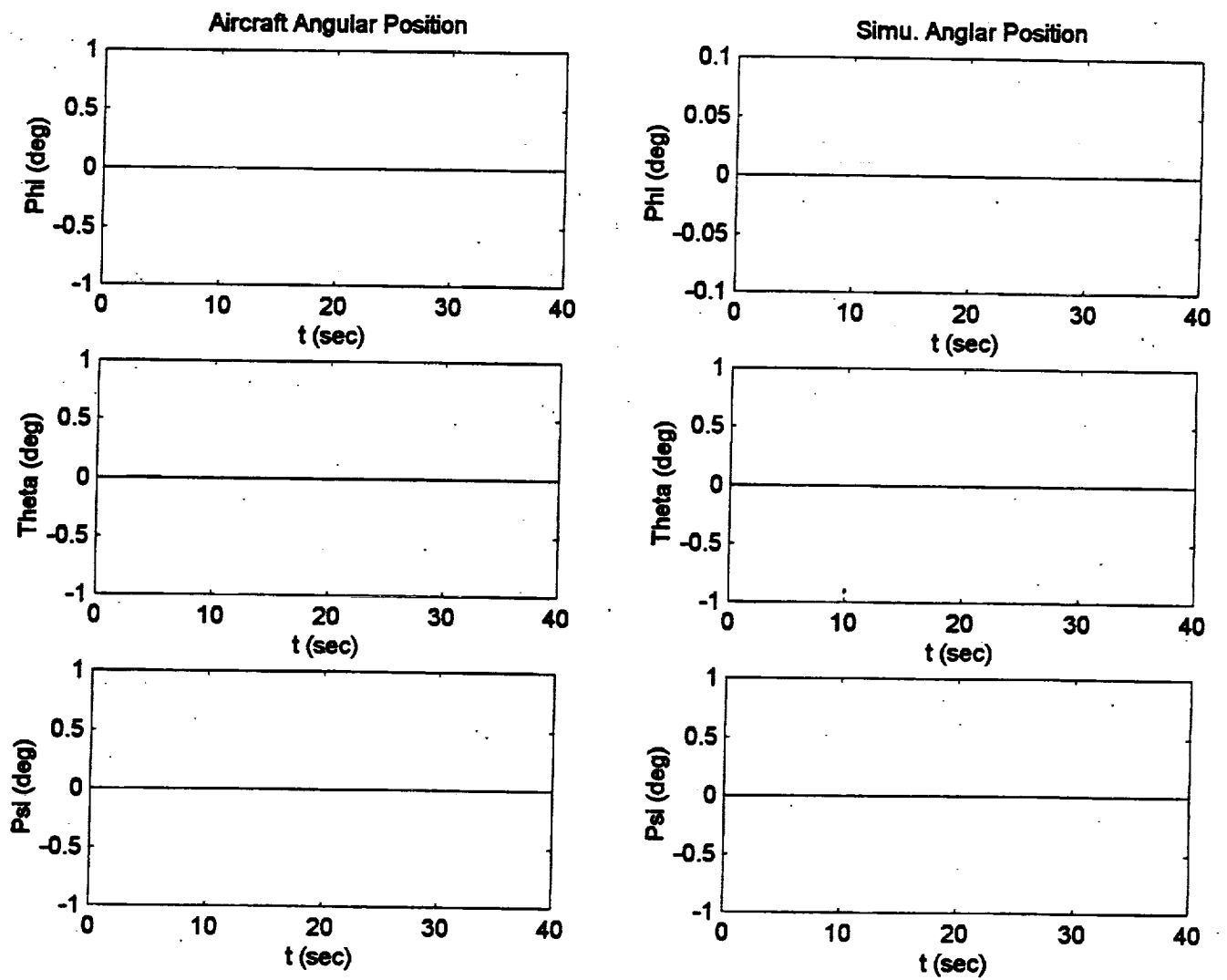


Figure C.13. Optimal Algorithm Heave Square Pulse Input.

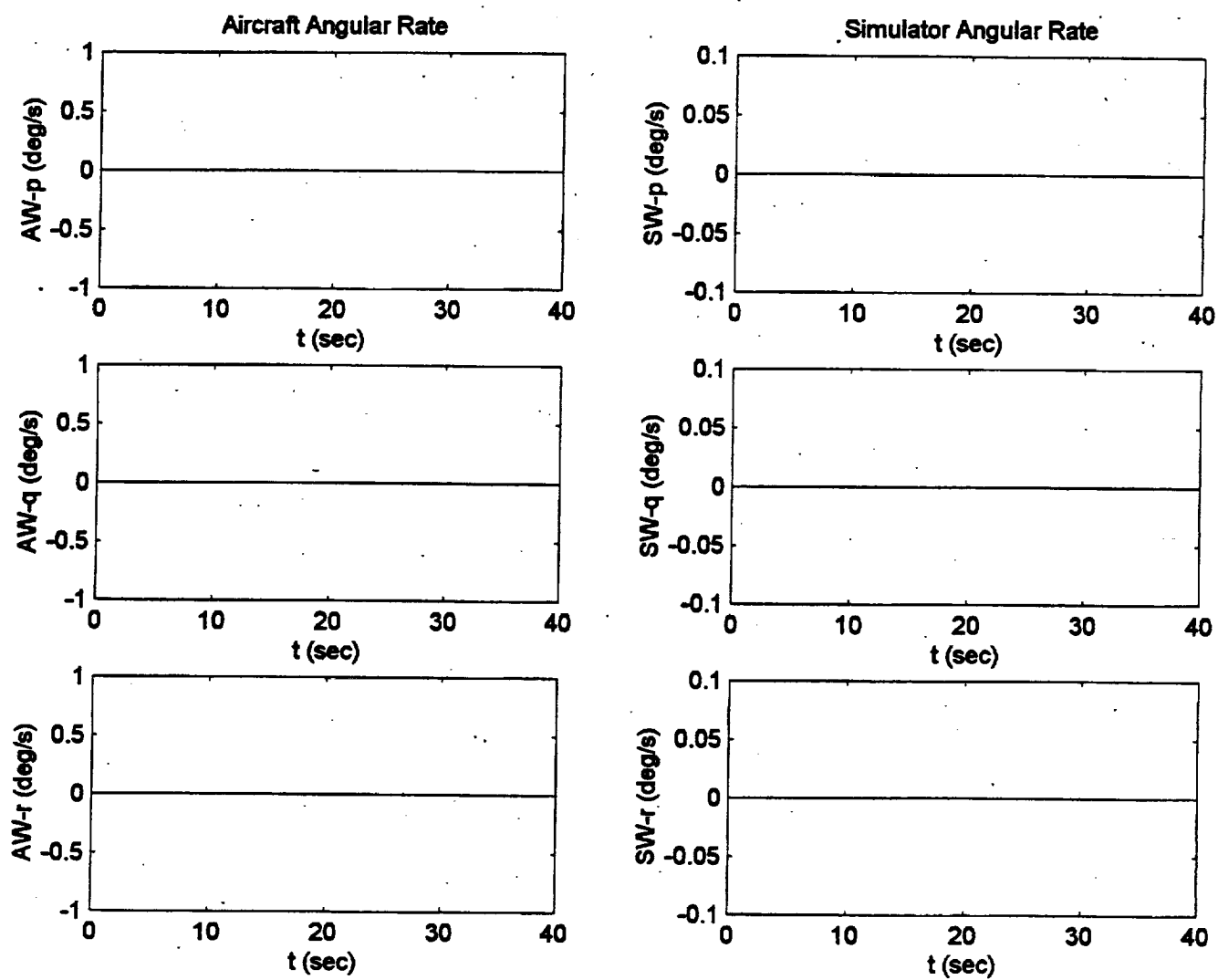


Figure C.13. Optimal Algorithm Heave Square Pulse Input.

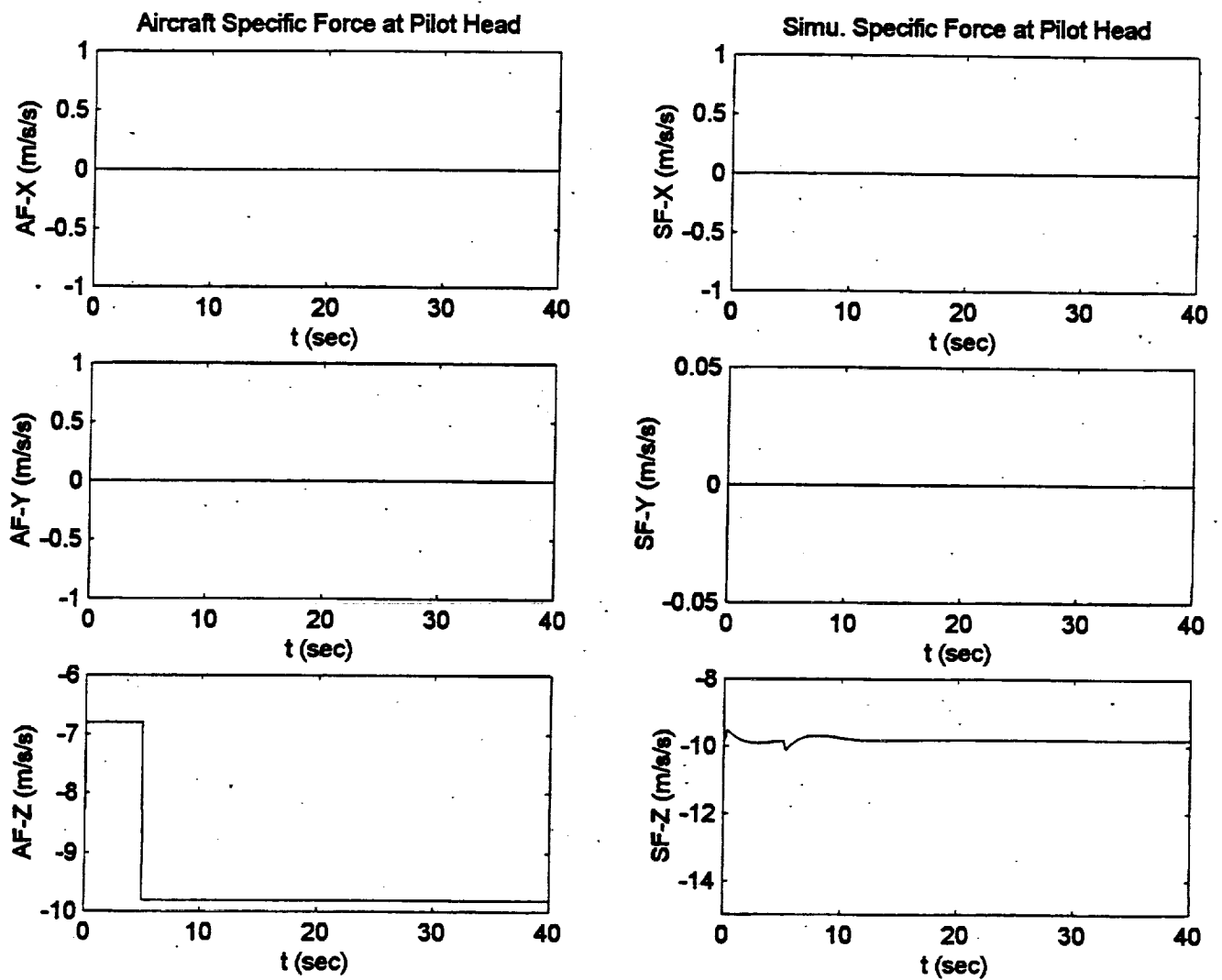


Figure C.13. Optimal Algorithm Heave Square Pulse Input.

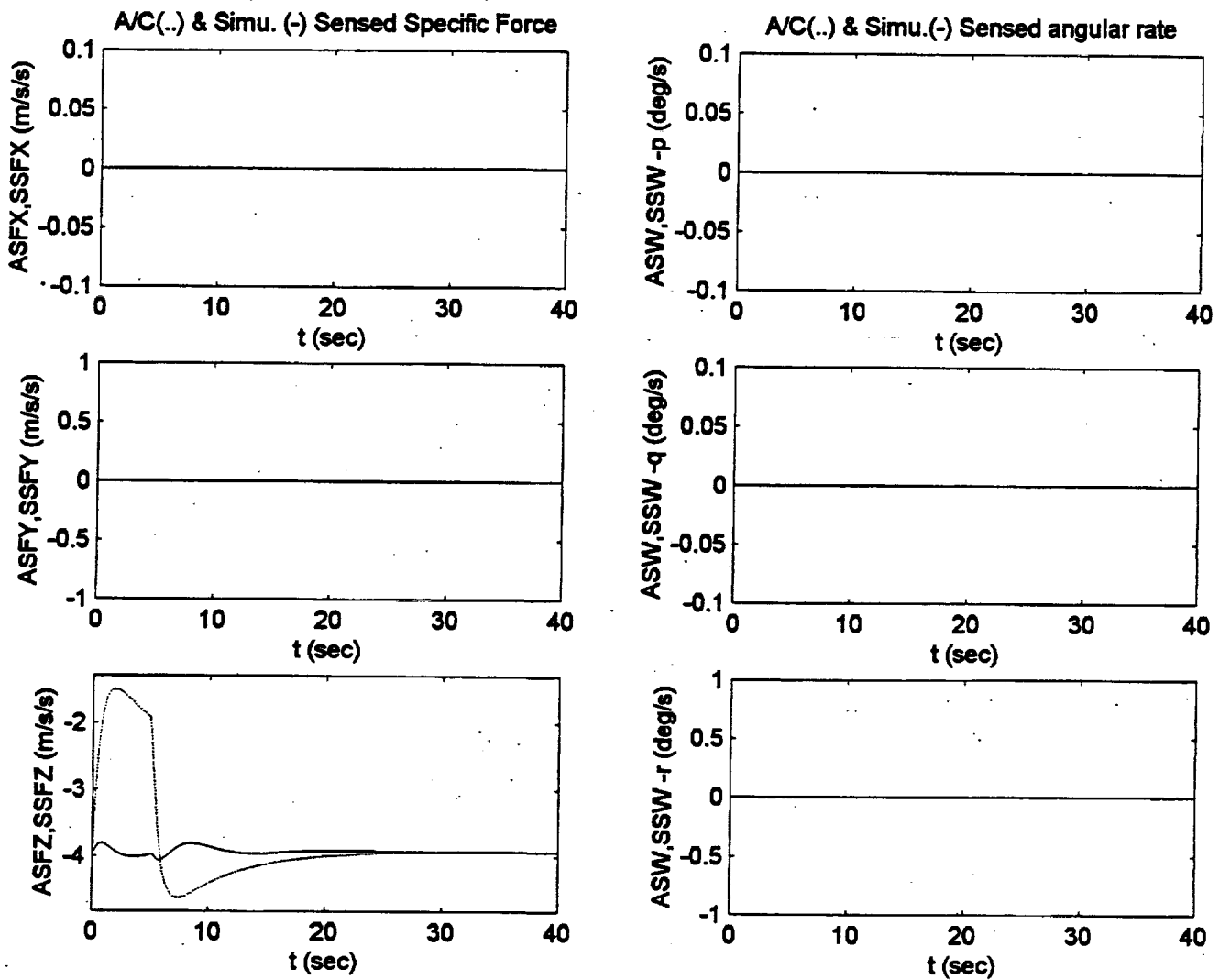


Figure C.13. Optimal Algorithm Heave Square Pulse Input.

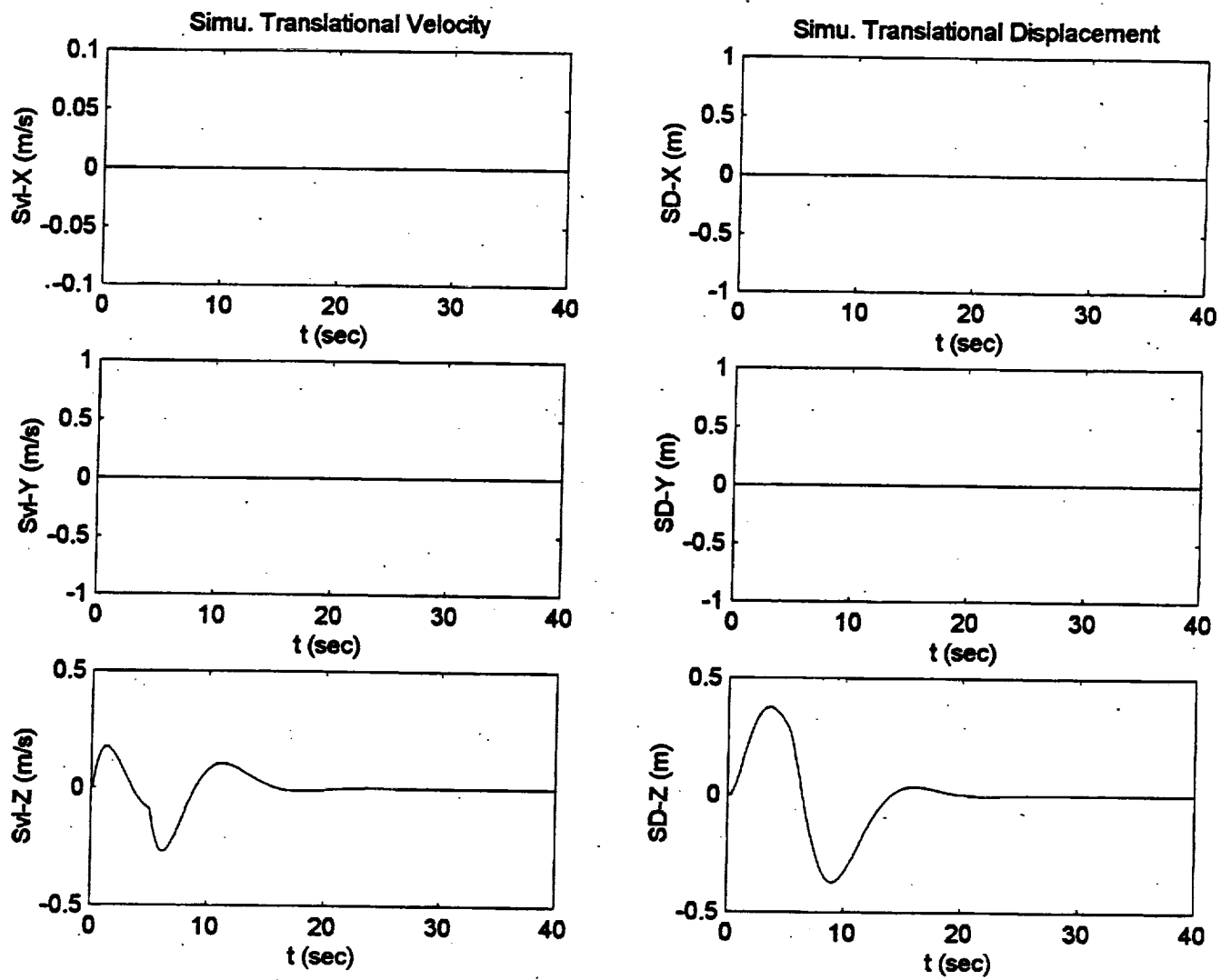


Figure C.13. Optimal Algorithm Heave Square Pulse Input.

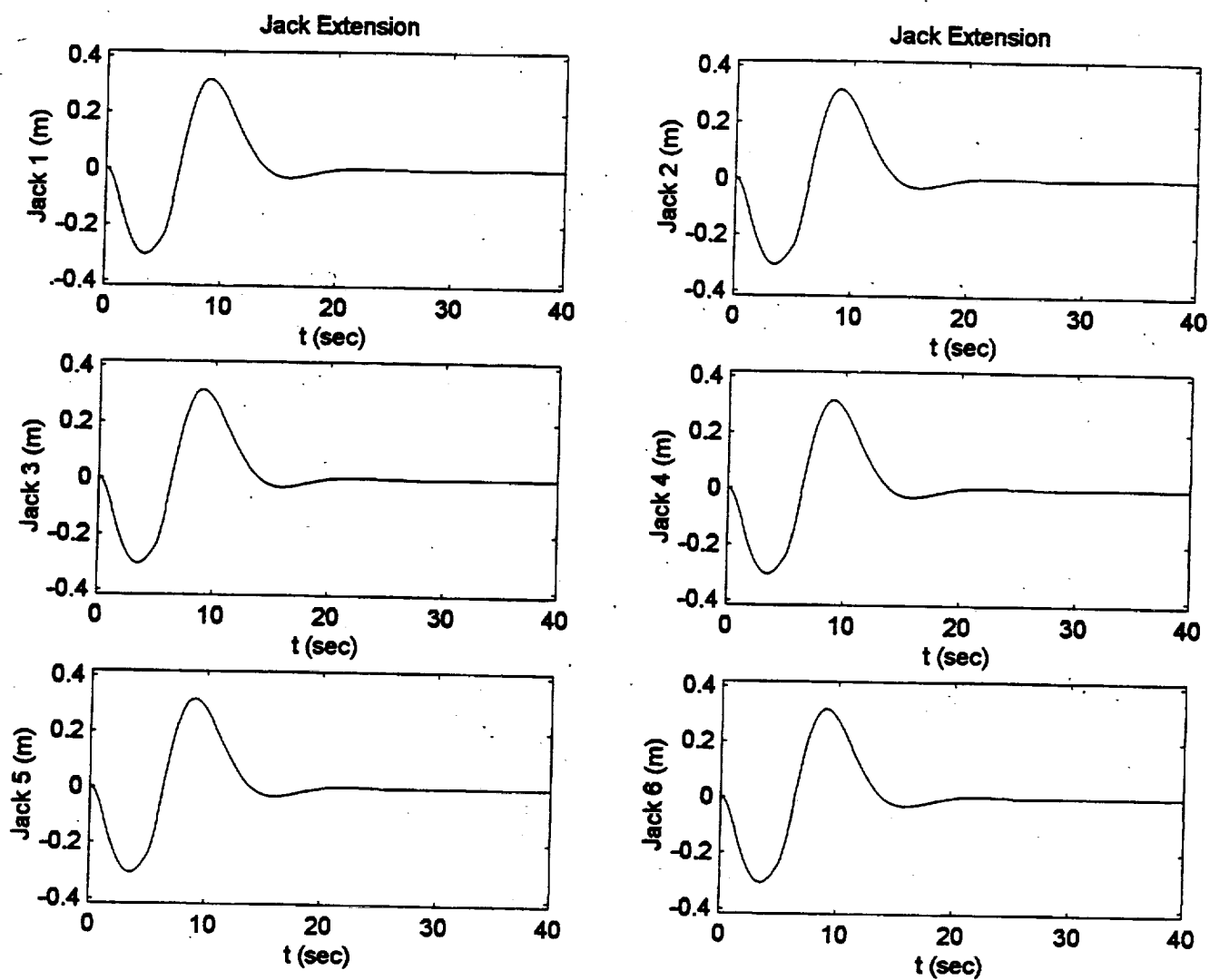


Figure C.13. Optimal Algorithm Heave Square Pulse Input.

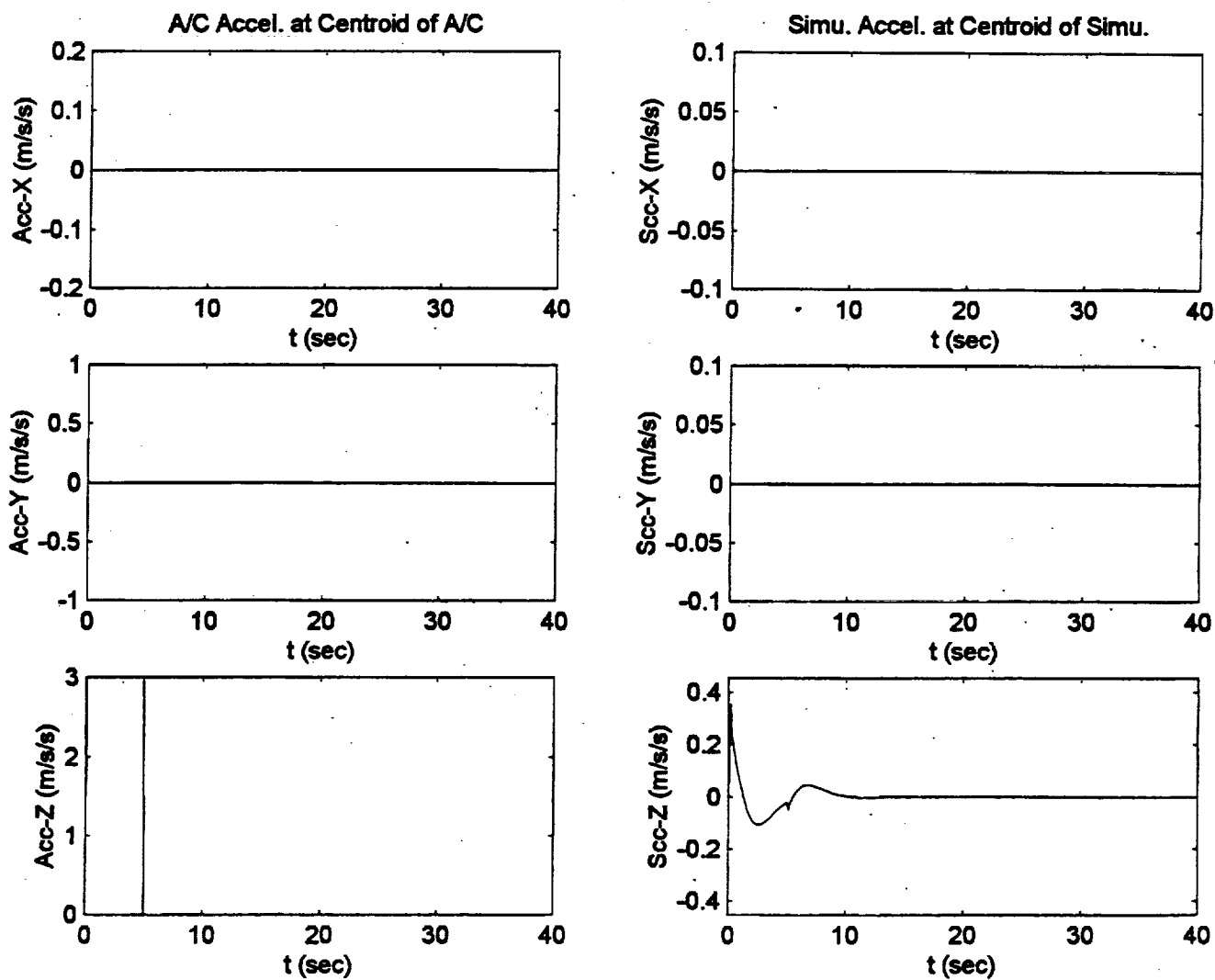


Figure C.14. NASA Adaptive Algorithm Heave Square Pulse Input.

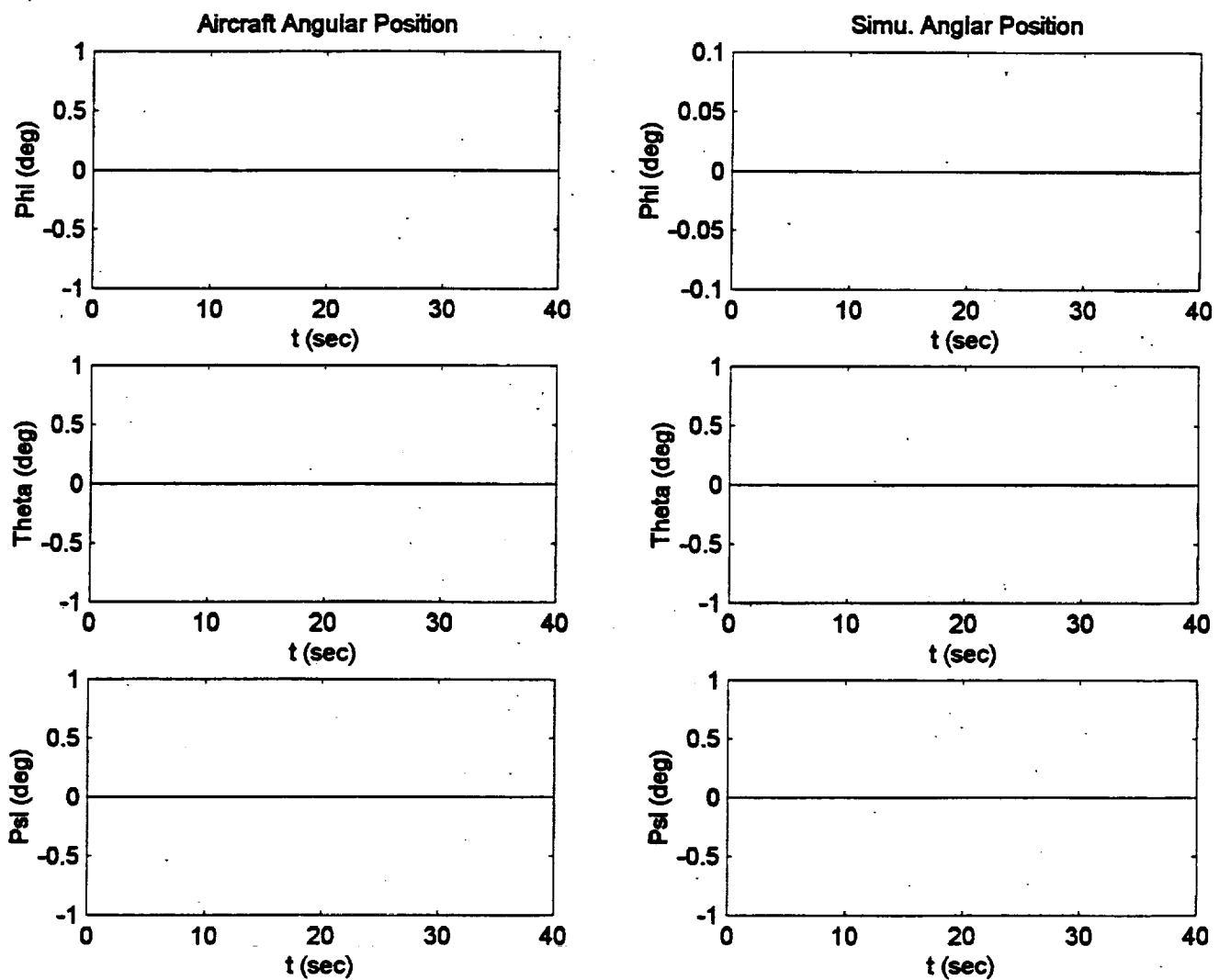


Figure C.14. NASA Adaptive Algorithm Heave Square Pulse Input.

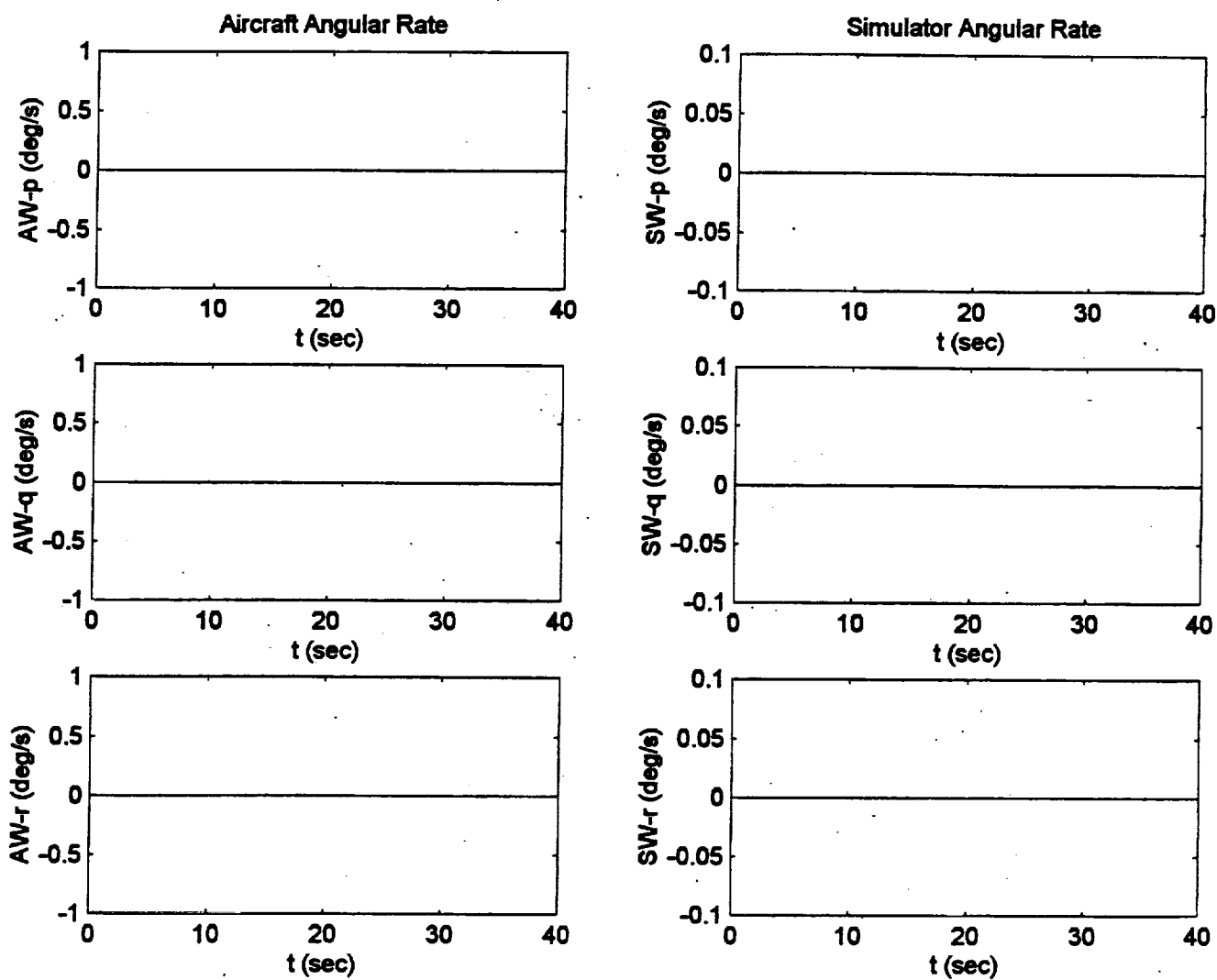


Figure C.14. NASA Adaptive Algorithm Heave Square Pulse Input.

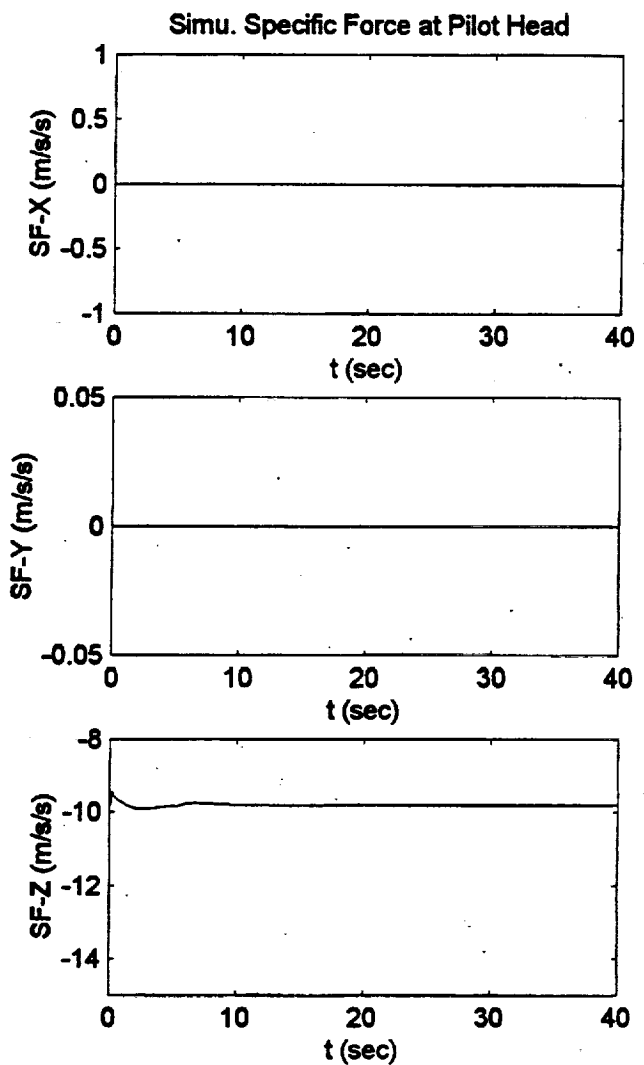
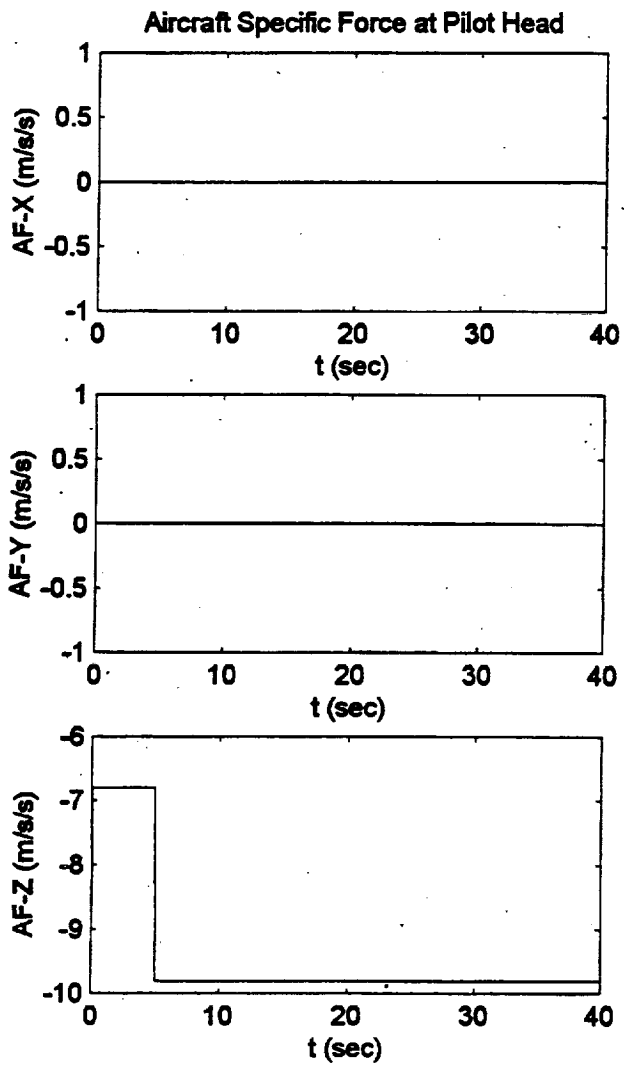


Figure C.14. NASA Adaptive Algorithm Heave Square Pulse Input.

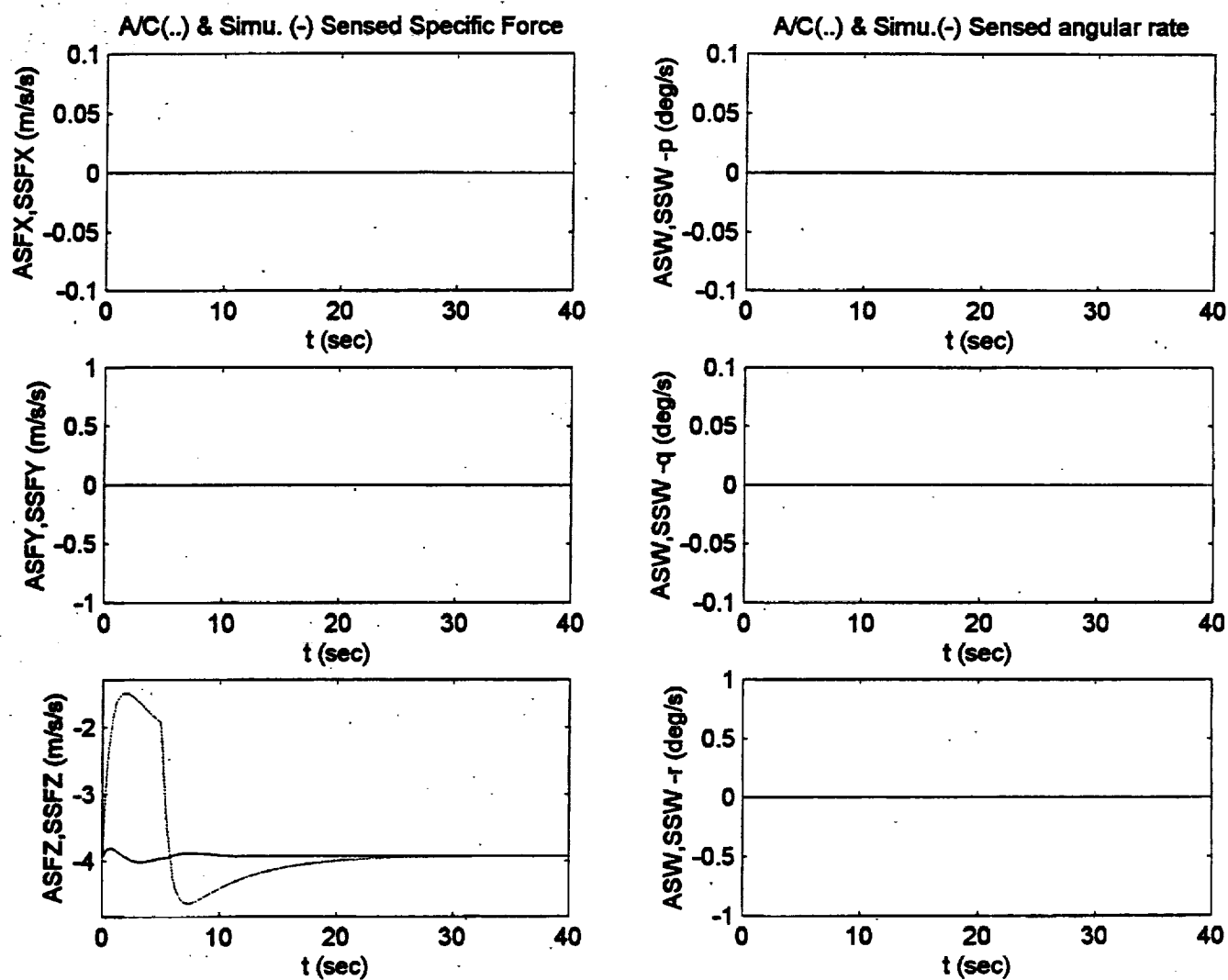


Figure C.14. NASA Adaptive Algorithm Heave Square Pulse Input.

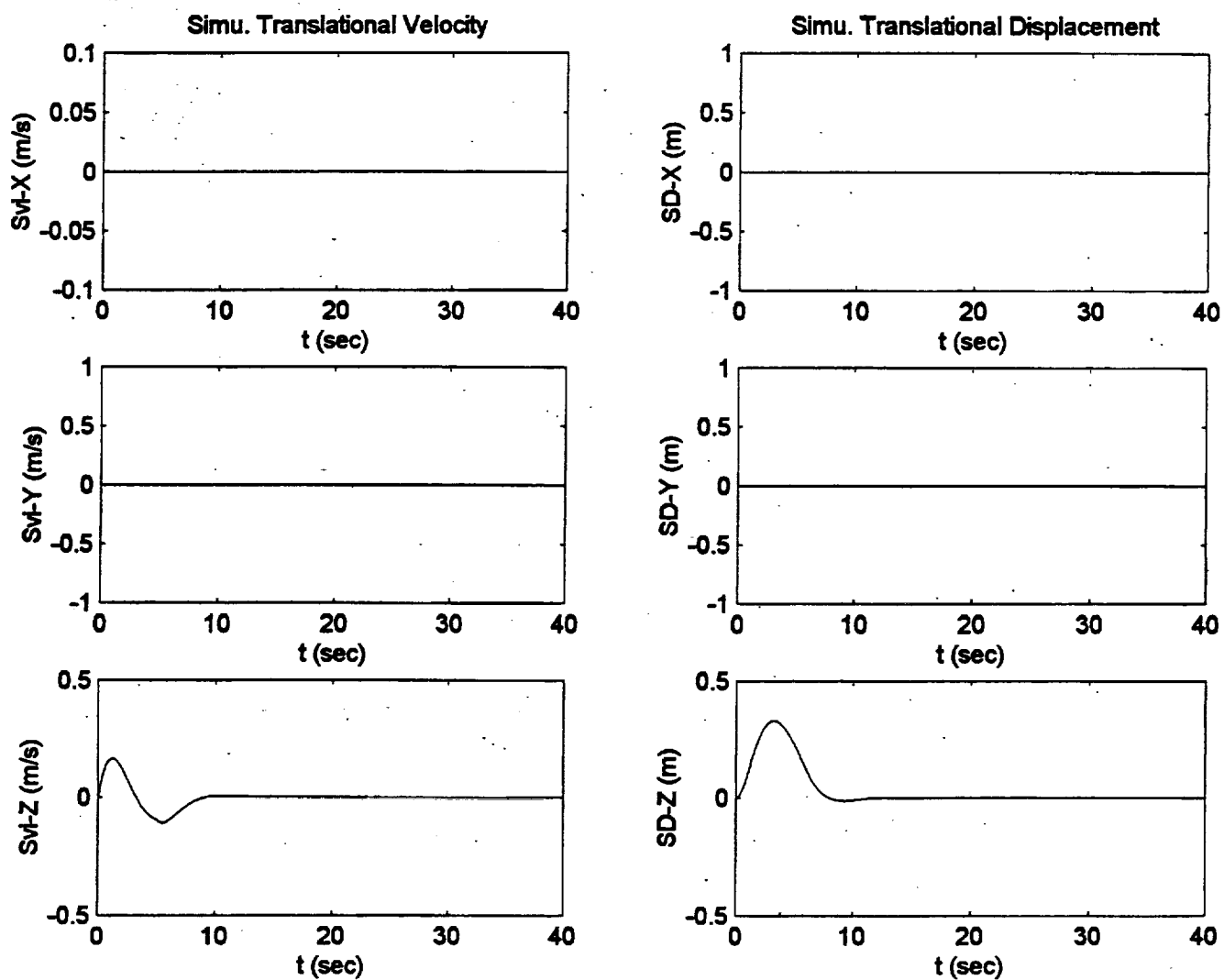


Figure C.14. NASA Adaptive Algorithm Heave Square Pulse Input.

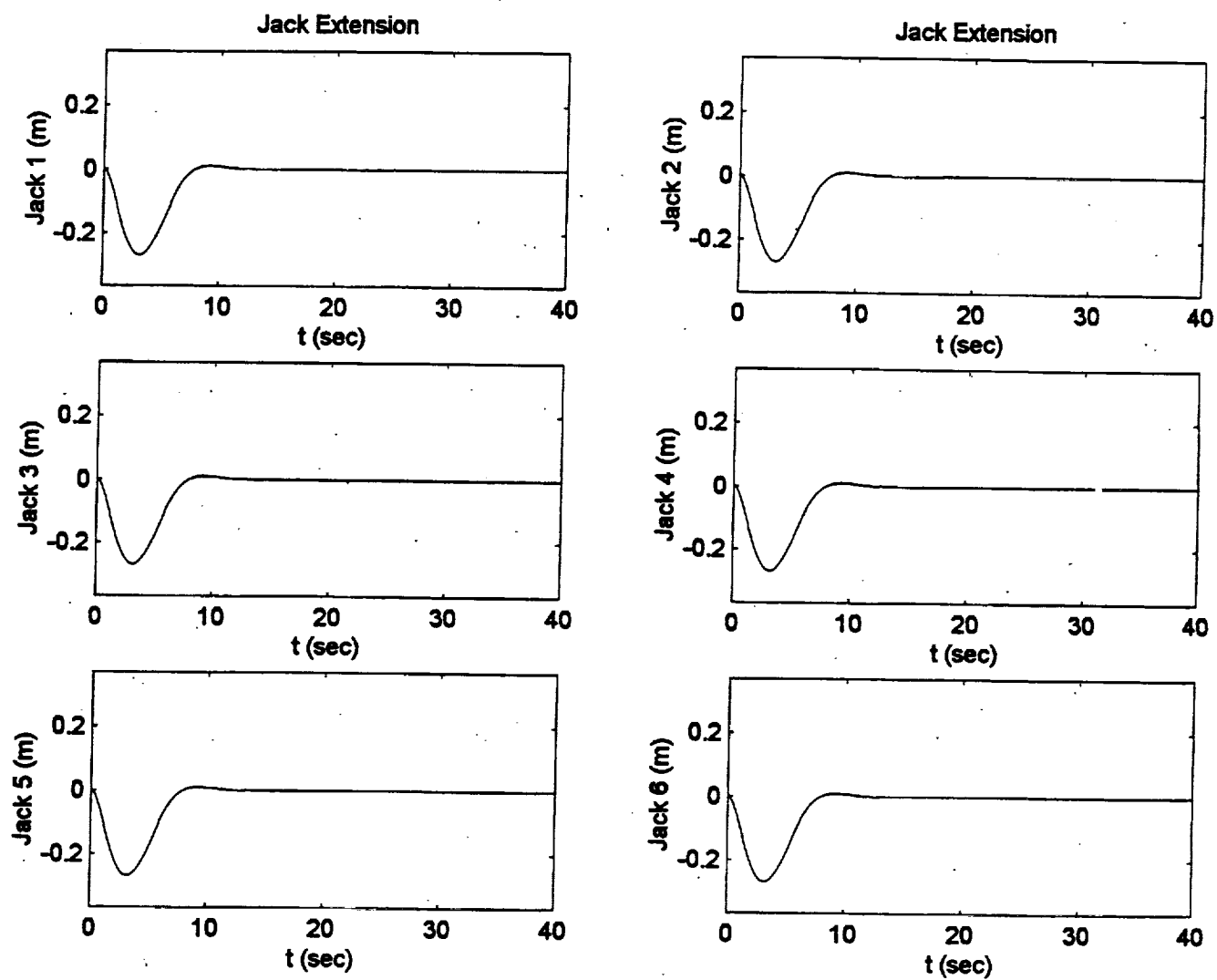


Figure C.14. NASA Adaptive Algorithm Heave Square Pulse Input.

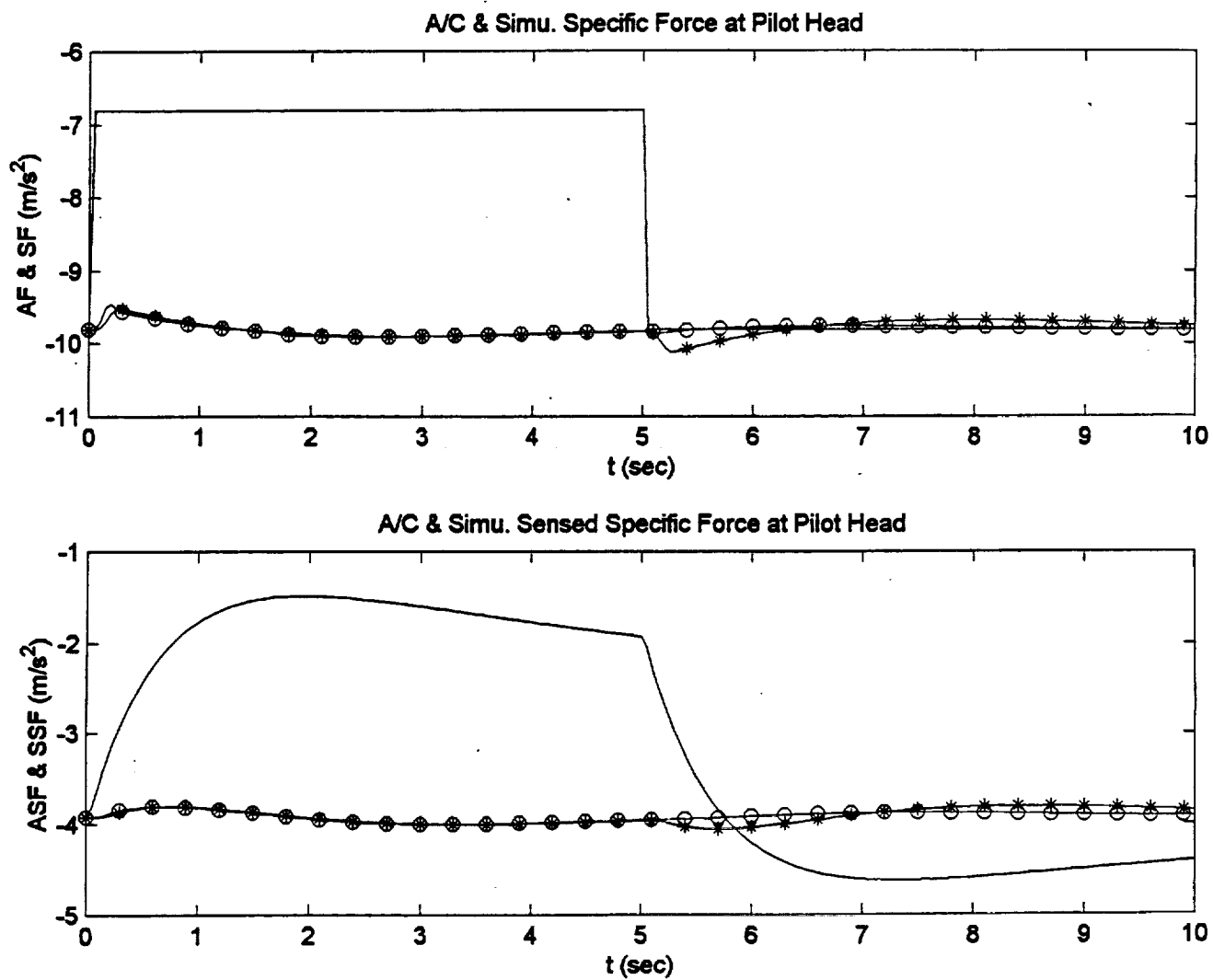


Figure C.15. Heave Square Pulse Input Specific Force Comparison.

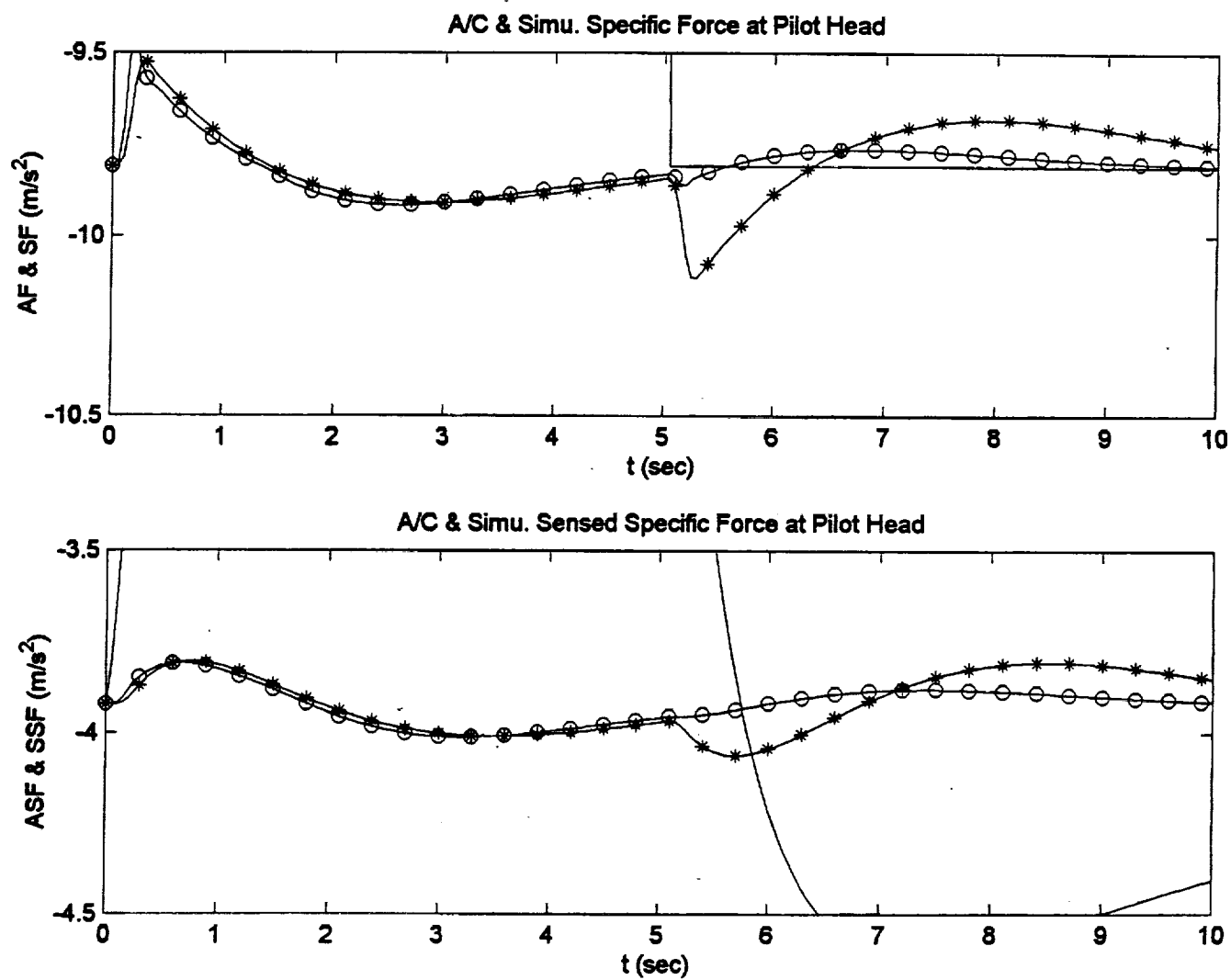


Figure C.15. Heave Square Pulse Input Specific Force Comparison.

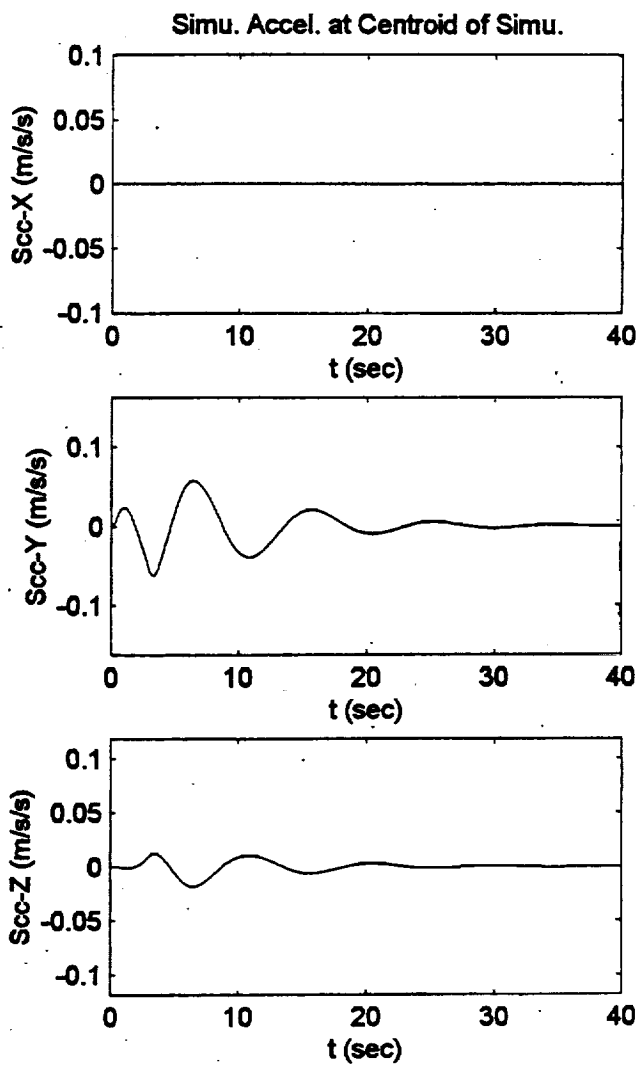
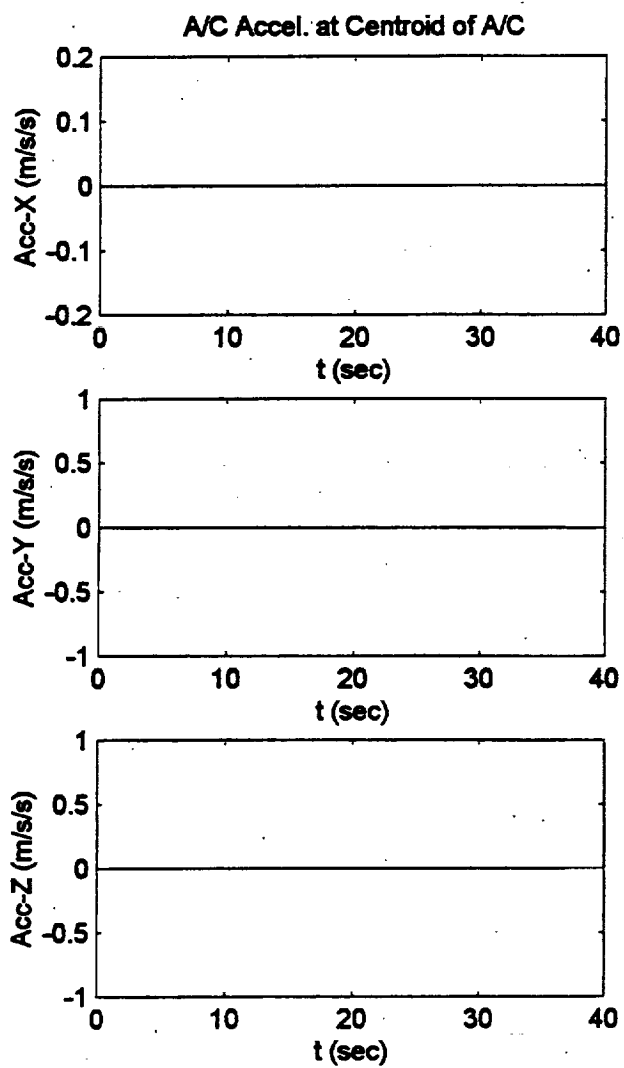


Figure C.16. Optimal Algorithm Roll Doublet Pulse Input.

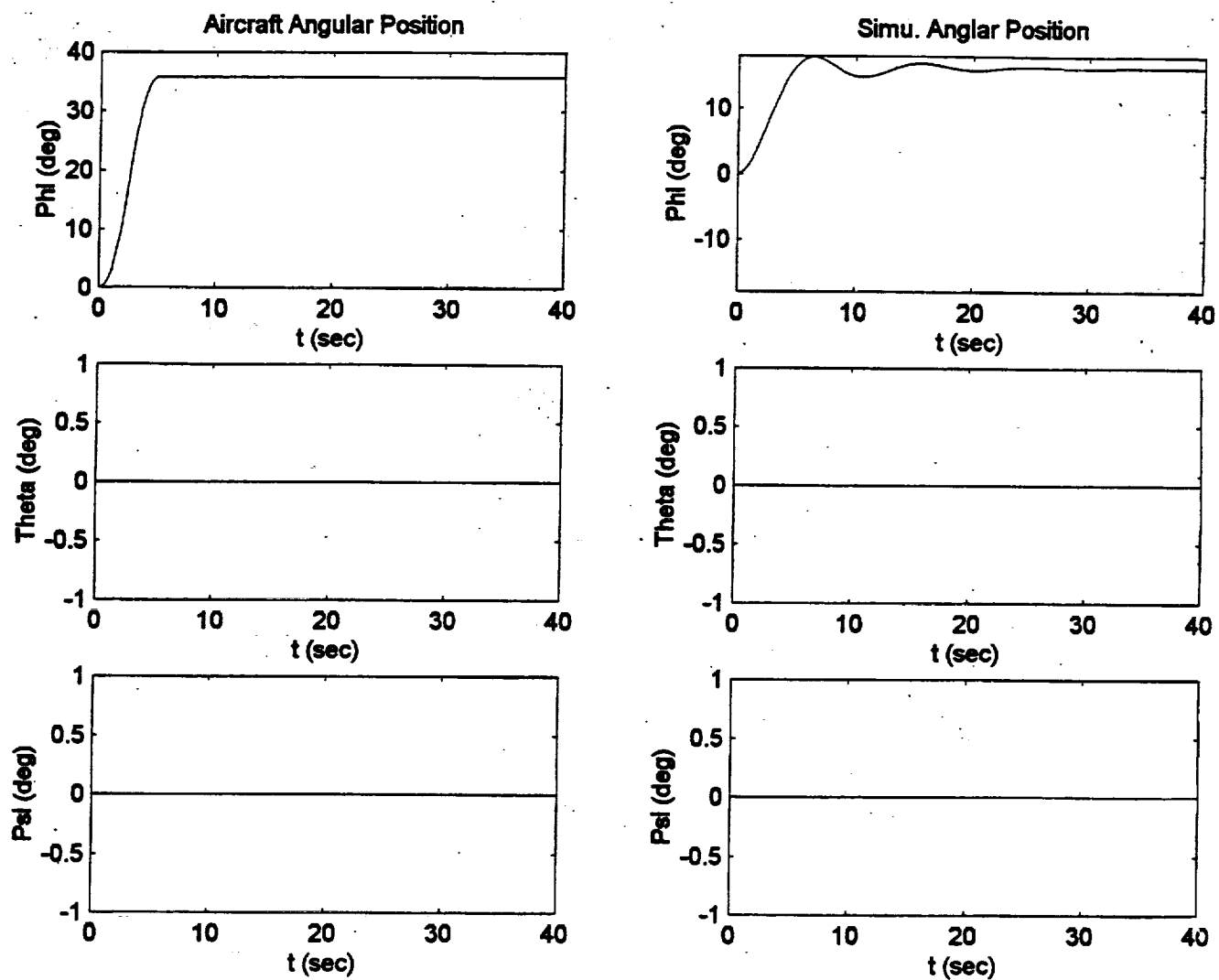


Figure C.16. Optimal Algorithm Roll Doublet Pulse Input.

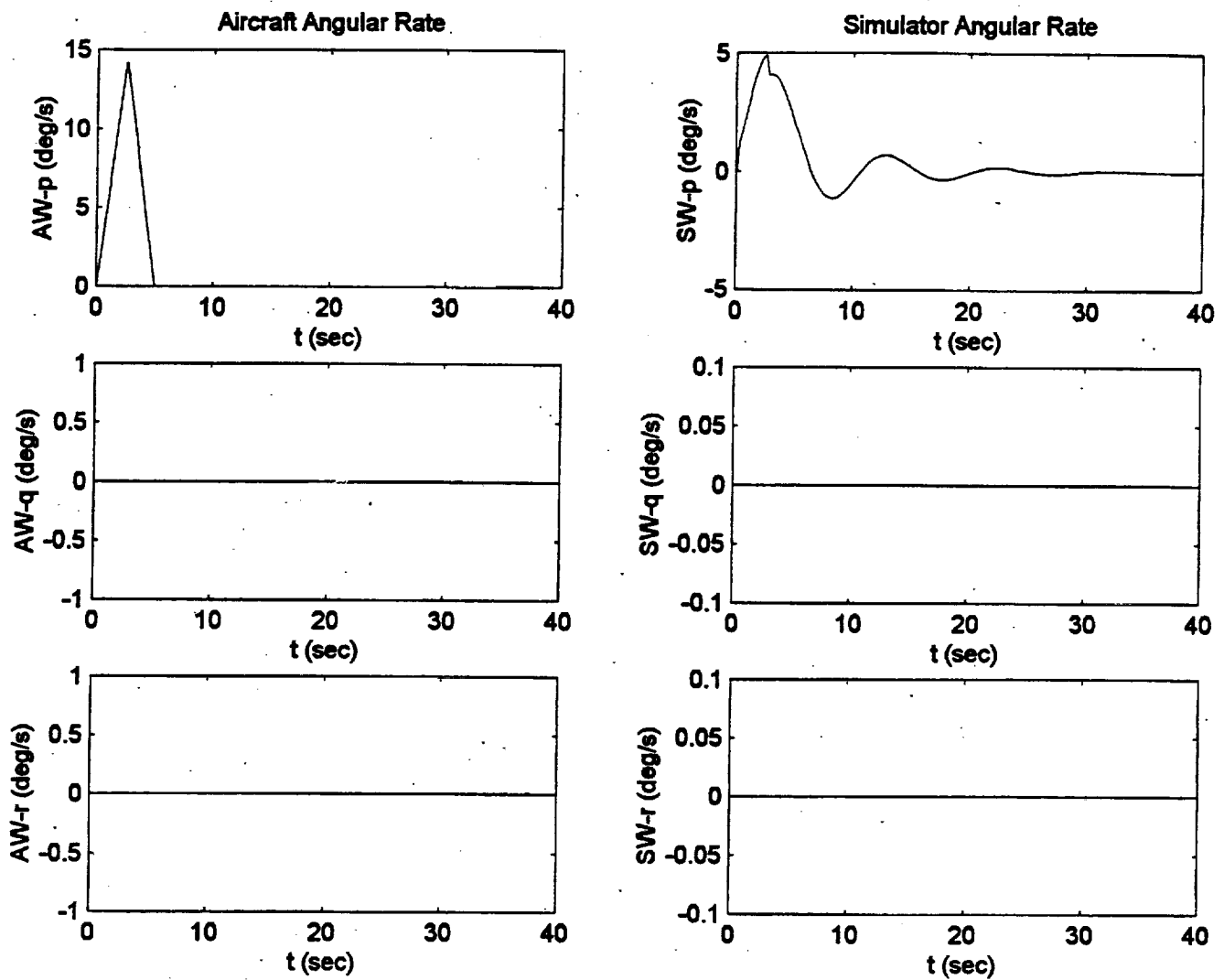


Figure C.16. Optimal Algorithm Roll Doublet Pulse Input.

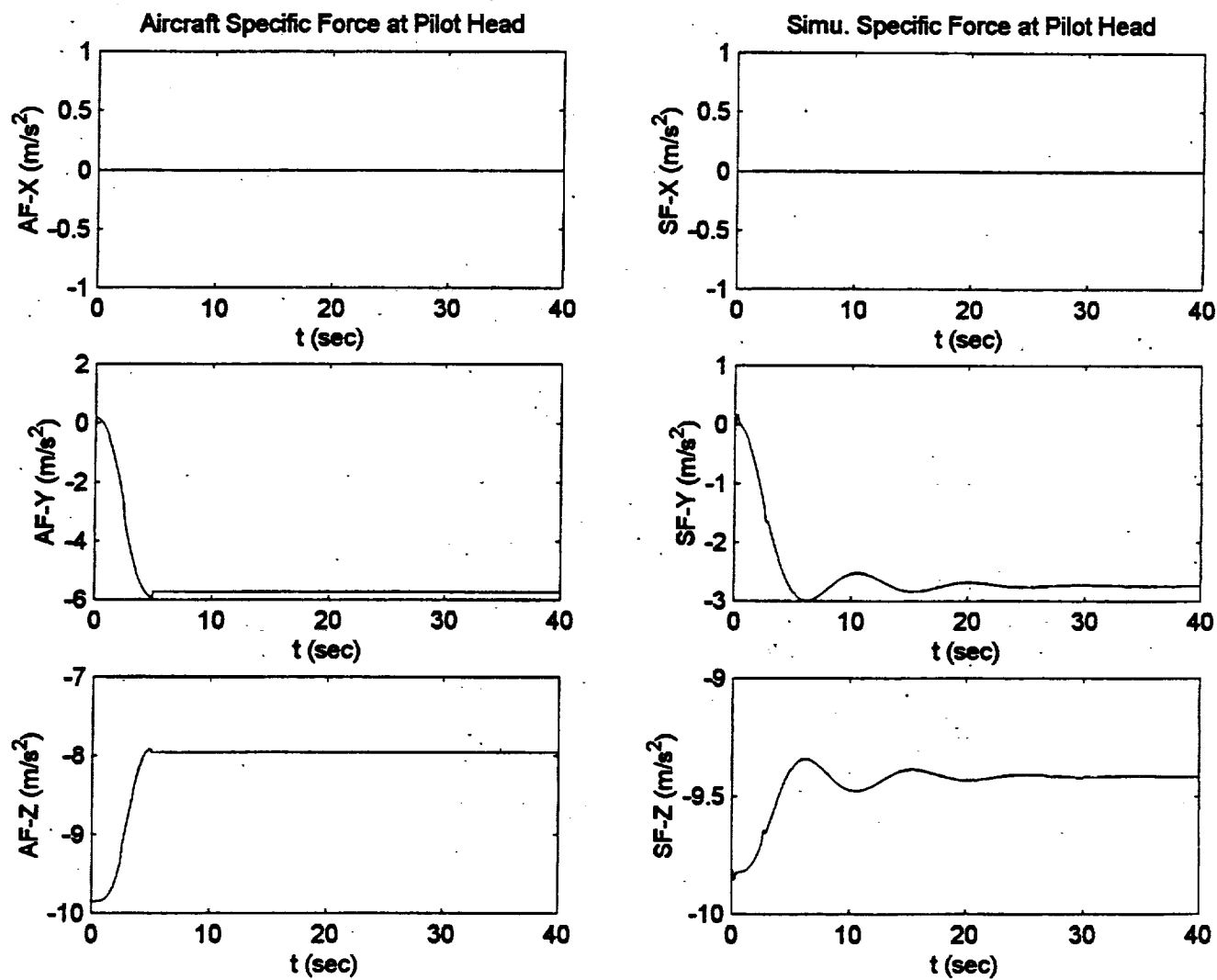


Figure C.16. Optimal Algorithm Roll Doublet Pulse Input.

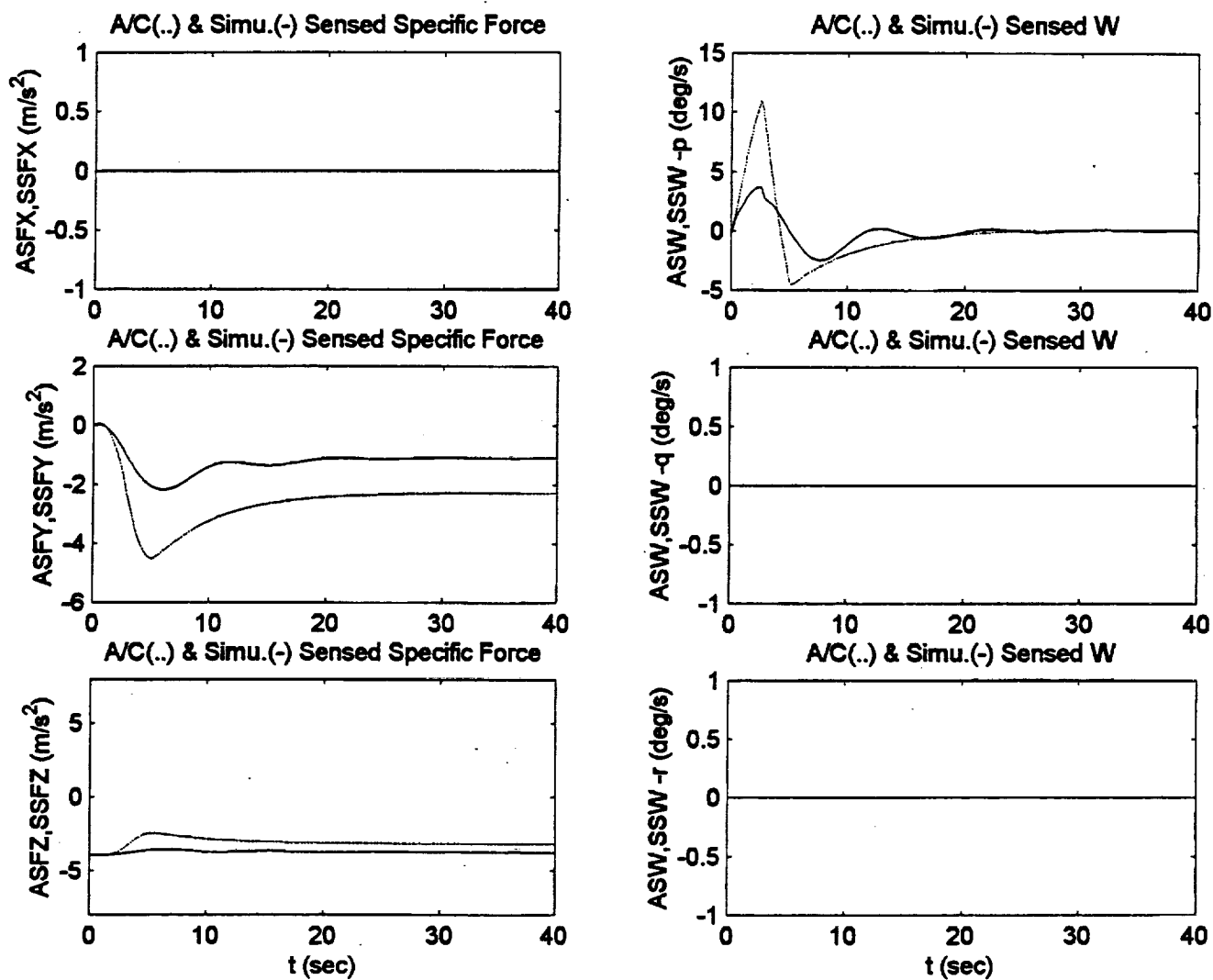


Figure C.16. Optimal Algorithm Roll Doublet Pulse Input.

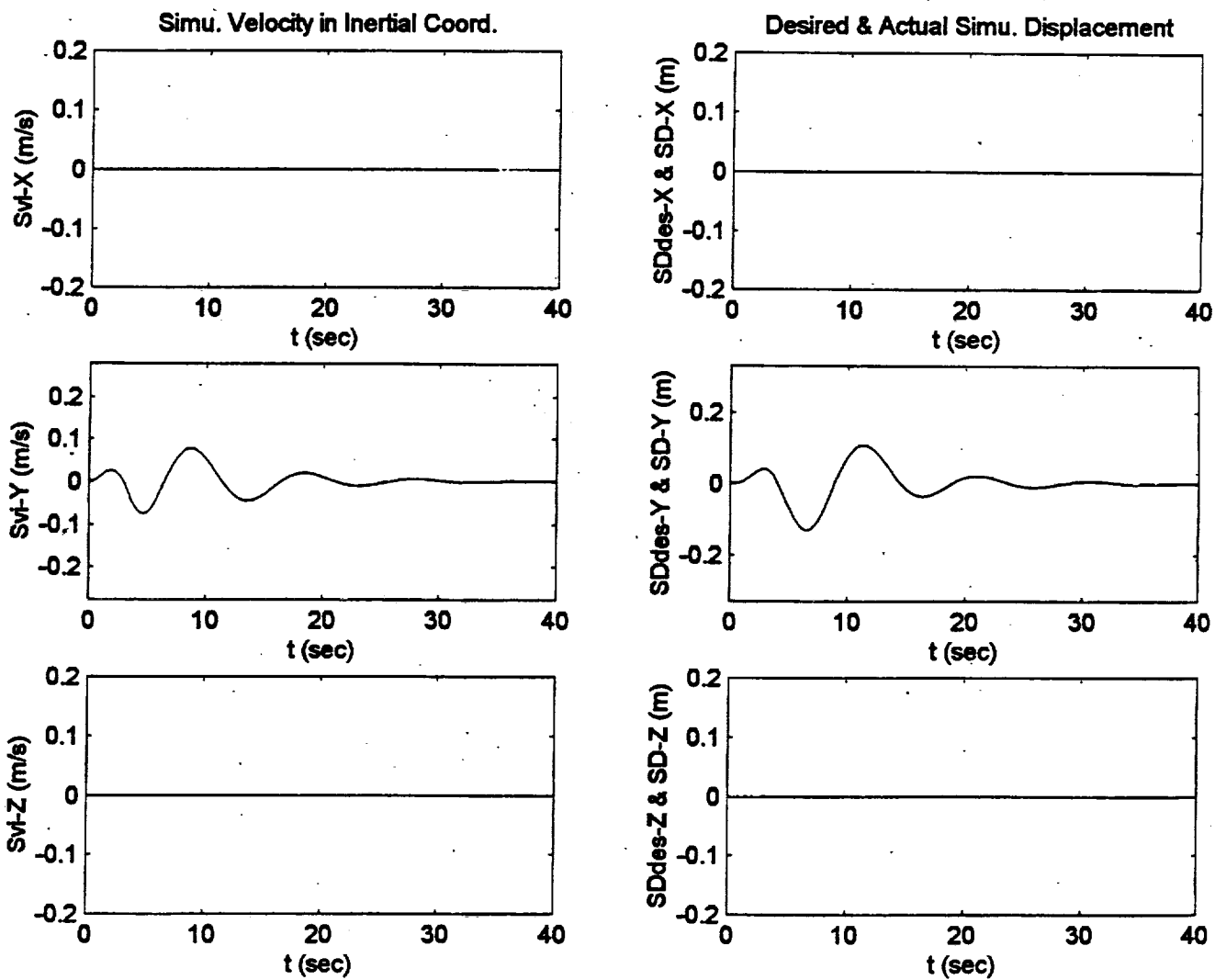


Figure C.16. Optimal Algorithm Roll Doublet Pulse Input.

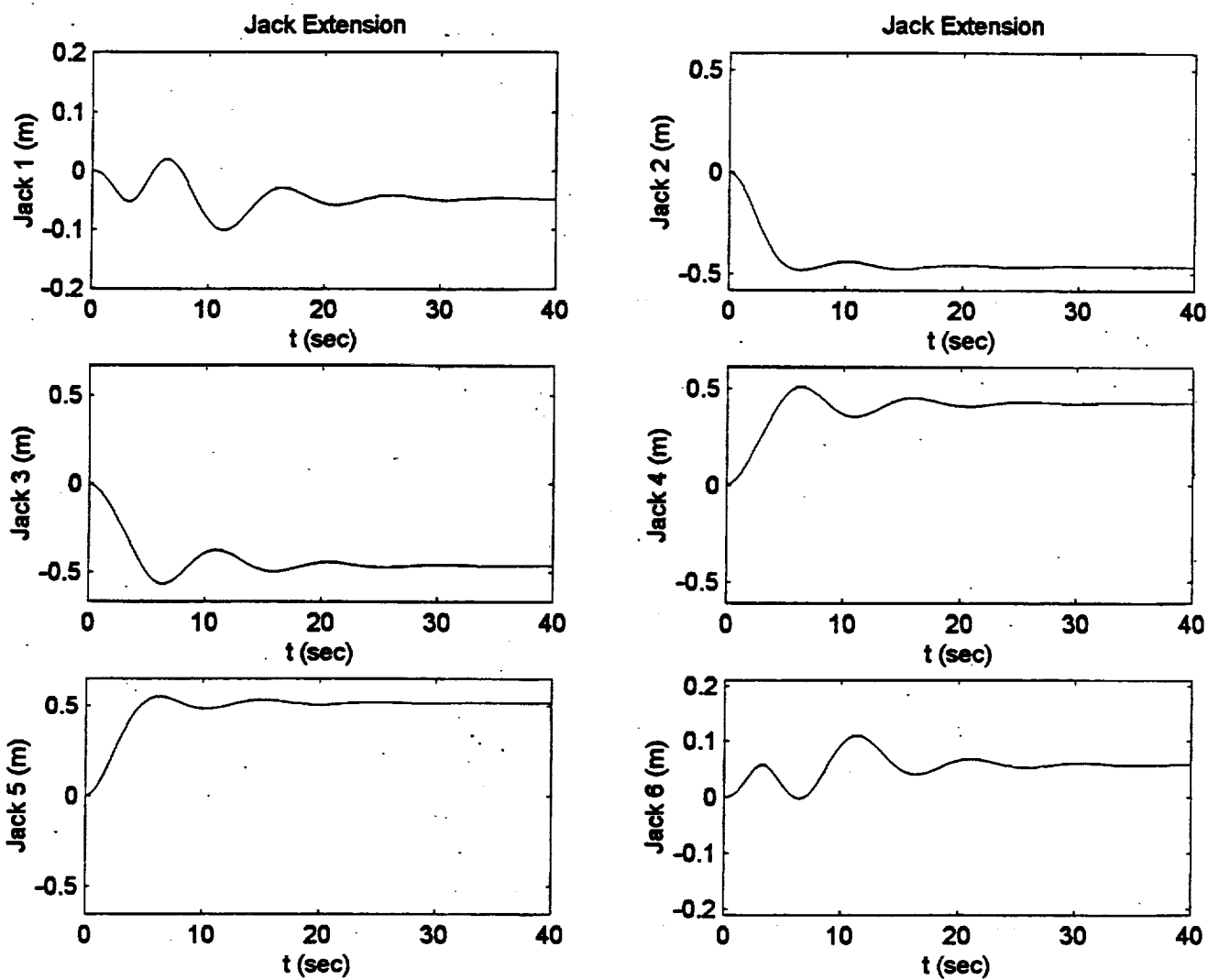


Figure C.16. Optimal Algorithm Roll Doublet Pulse Input.

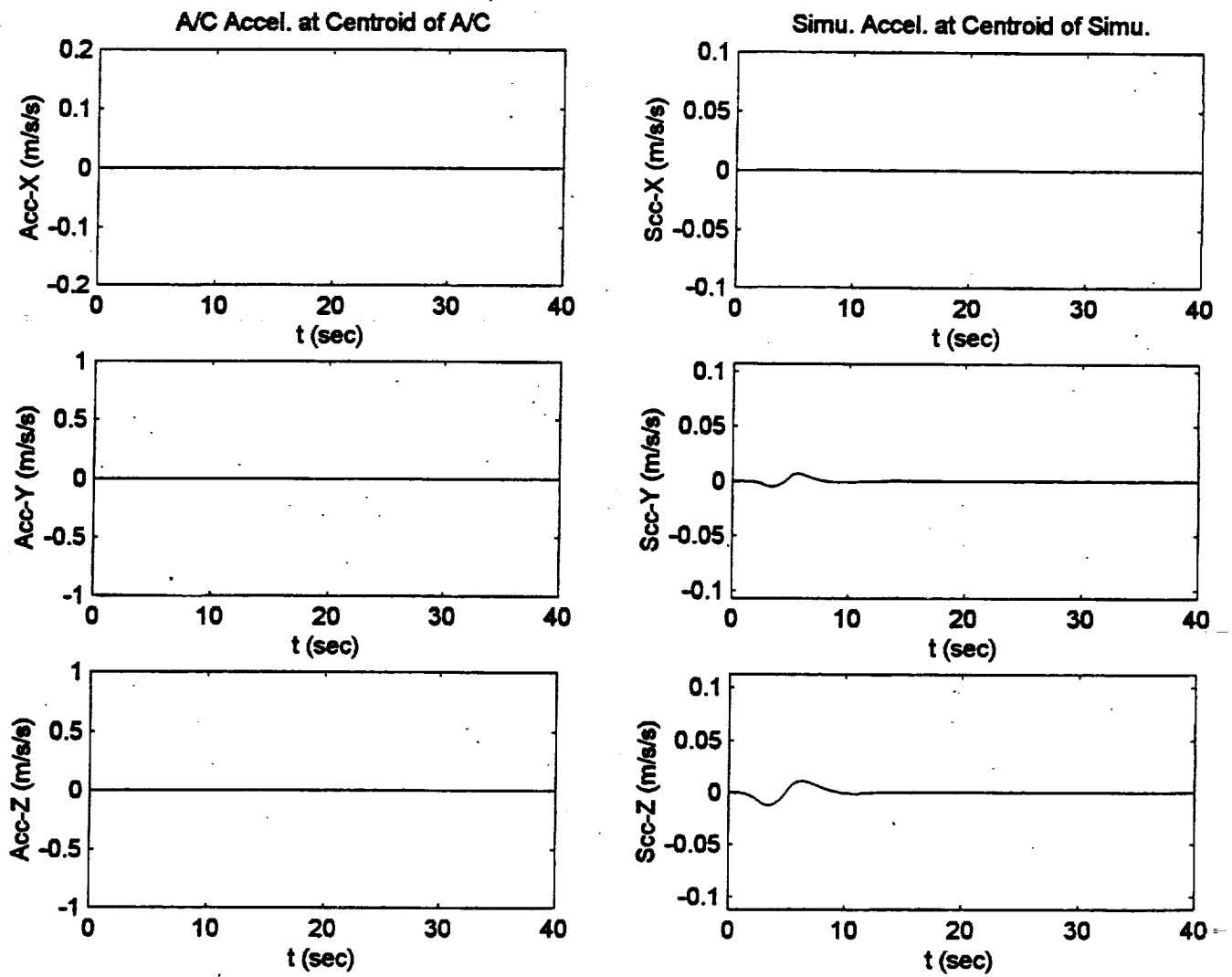


Figure C.17. NASA Adaptive Algorithm Roll Doublet Pulse Input.

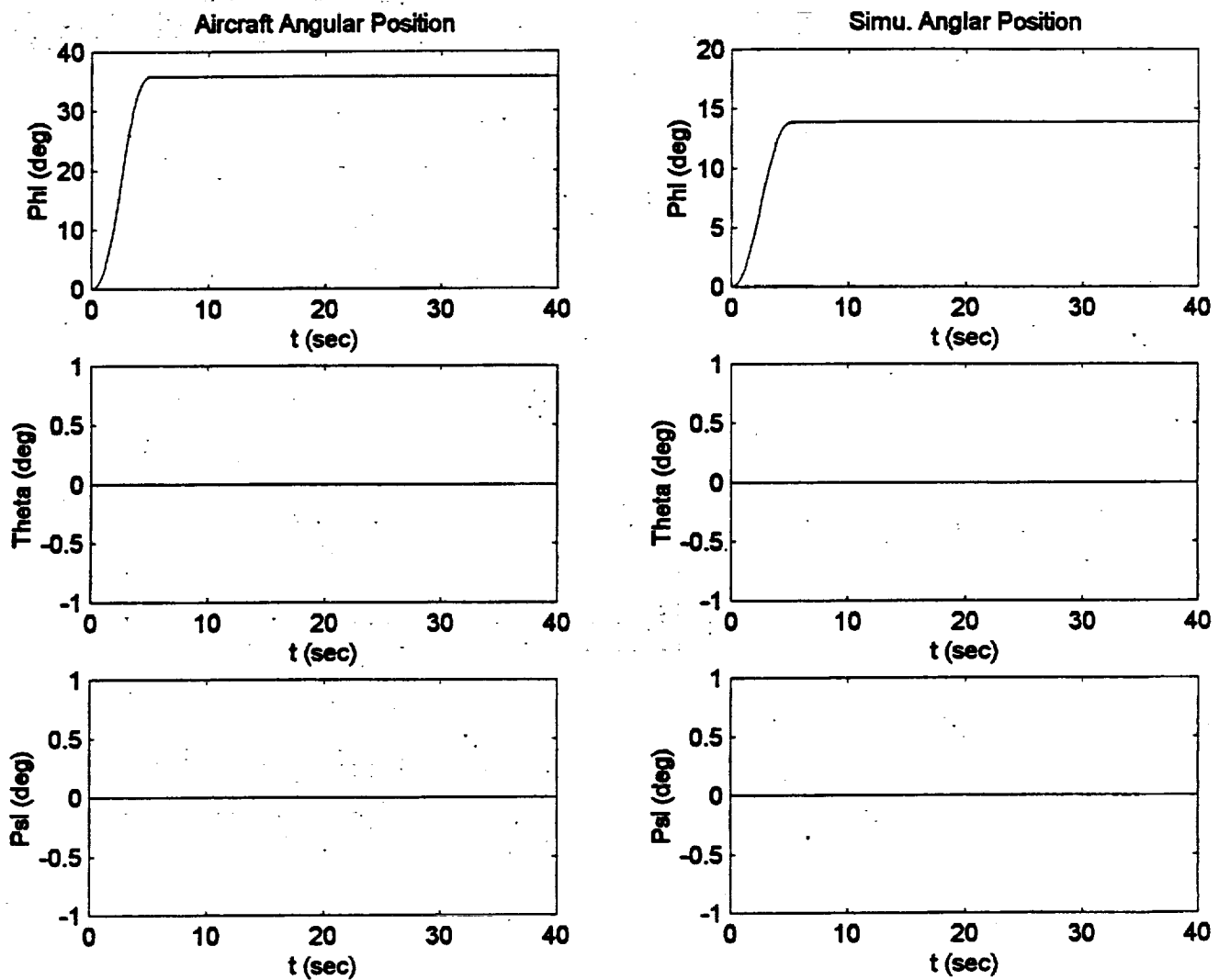


Figure C.17. NASA Adaptive Algorithm Roll Doublet Pulse Input.

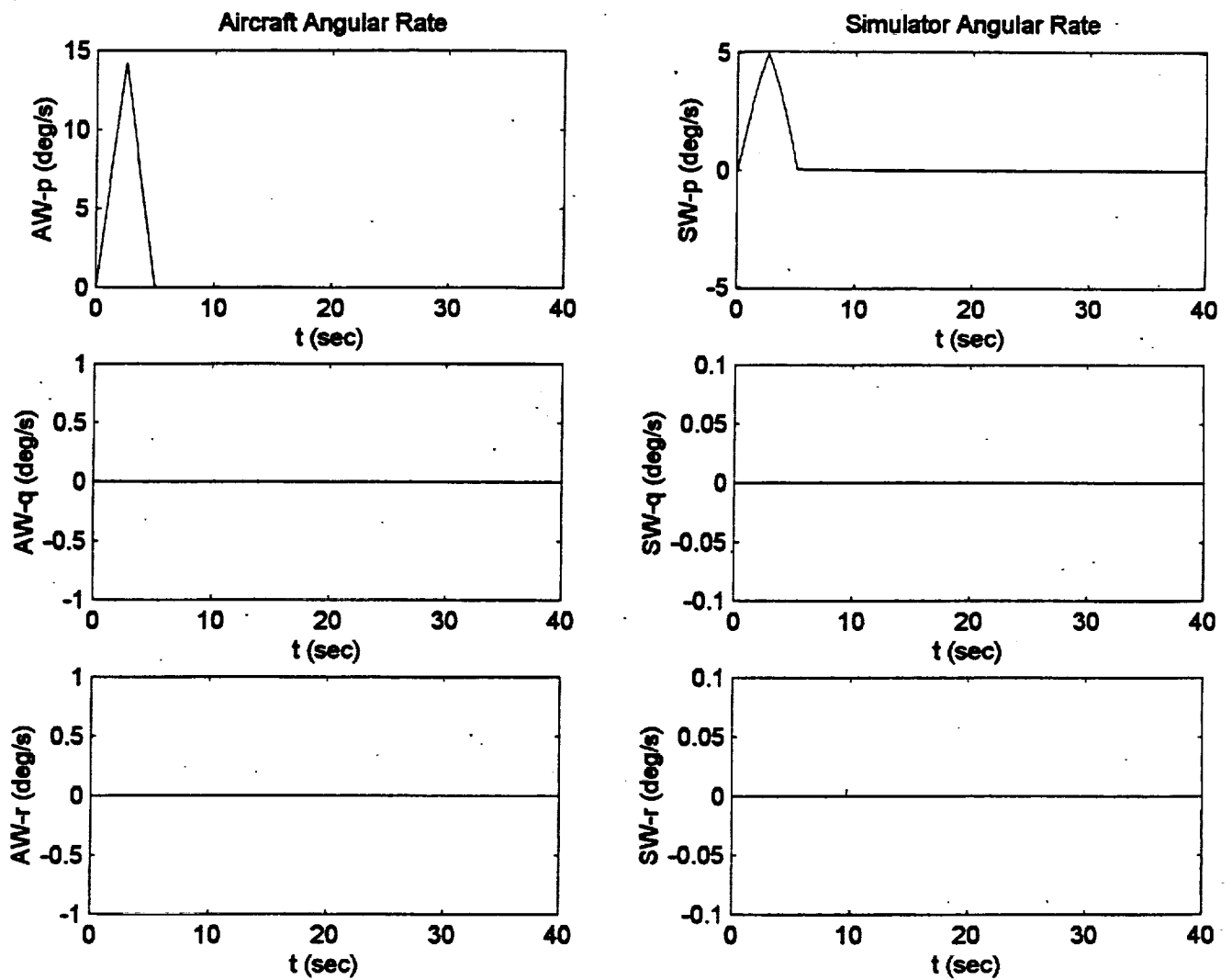


Figure C.17. NASA Adaptive Algorithm Roll Doublet Pulse Input.

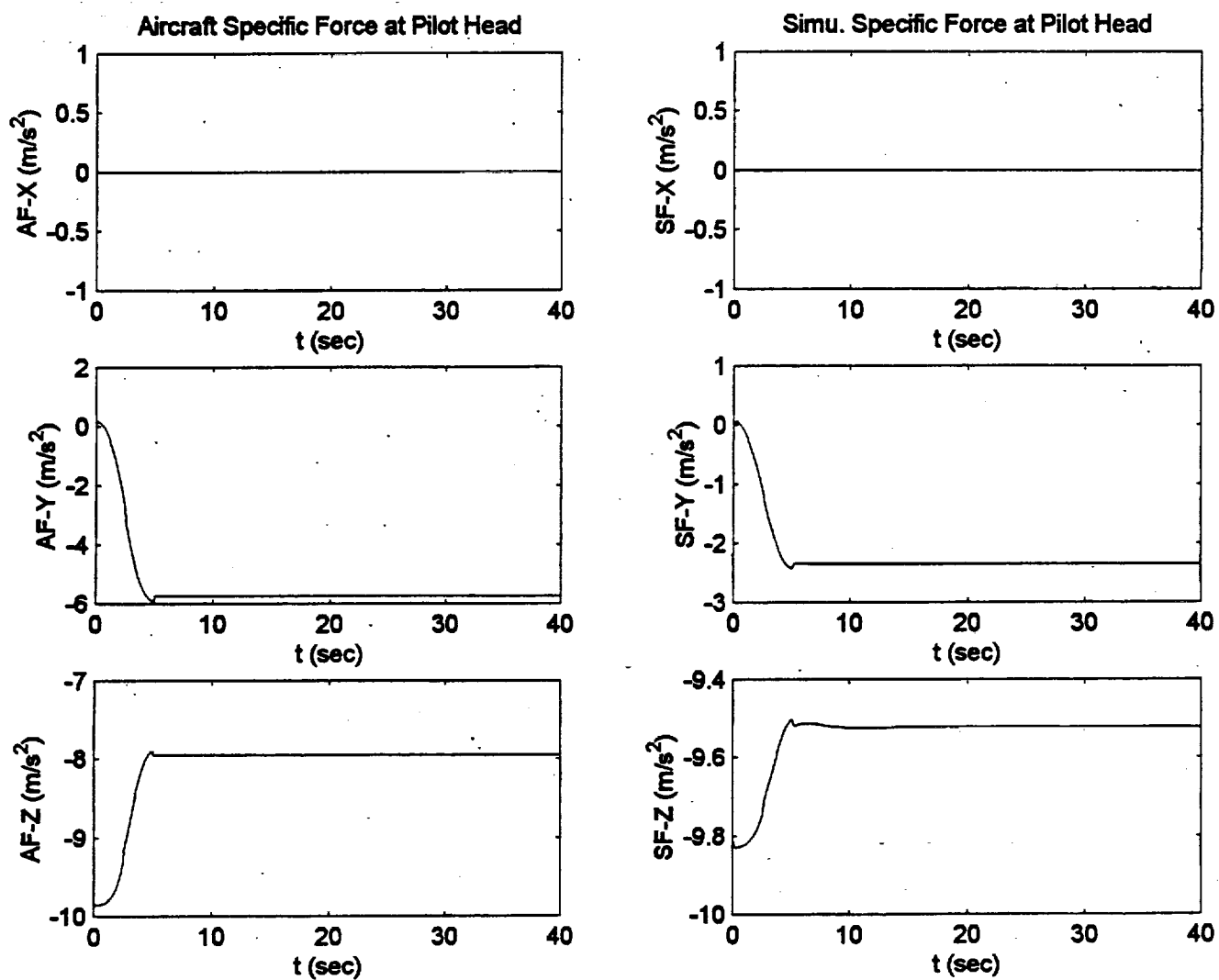


Figure C.17. NASA Adaptive Algorithm Roll Doublet Pulse Input.

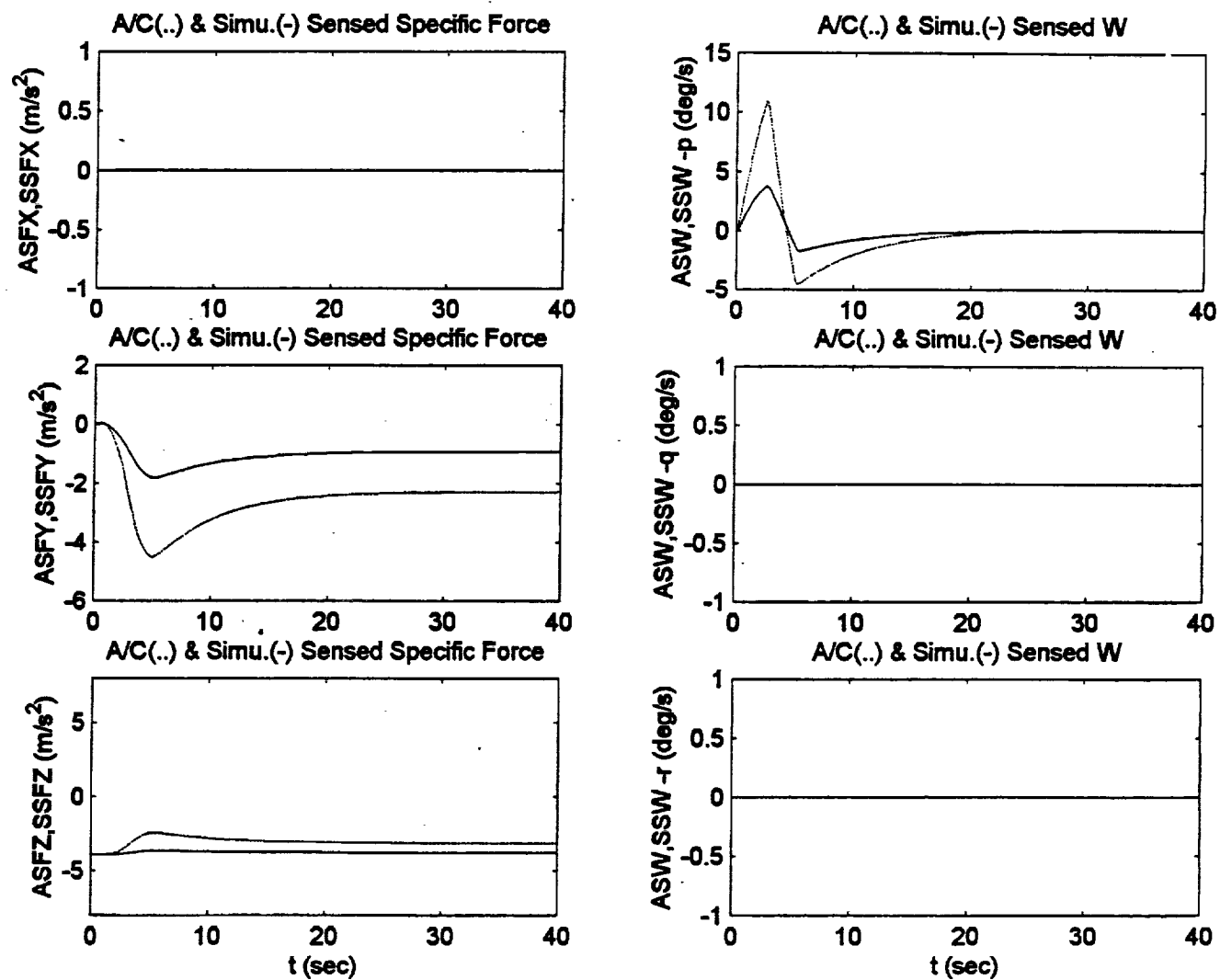


Figure C.17. NASA Adaptive Algorithm Roll Doublet Pulse Input.

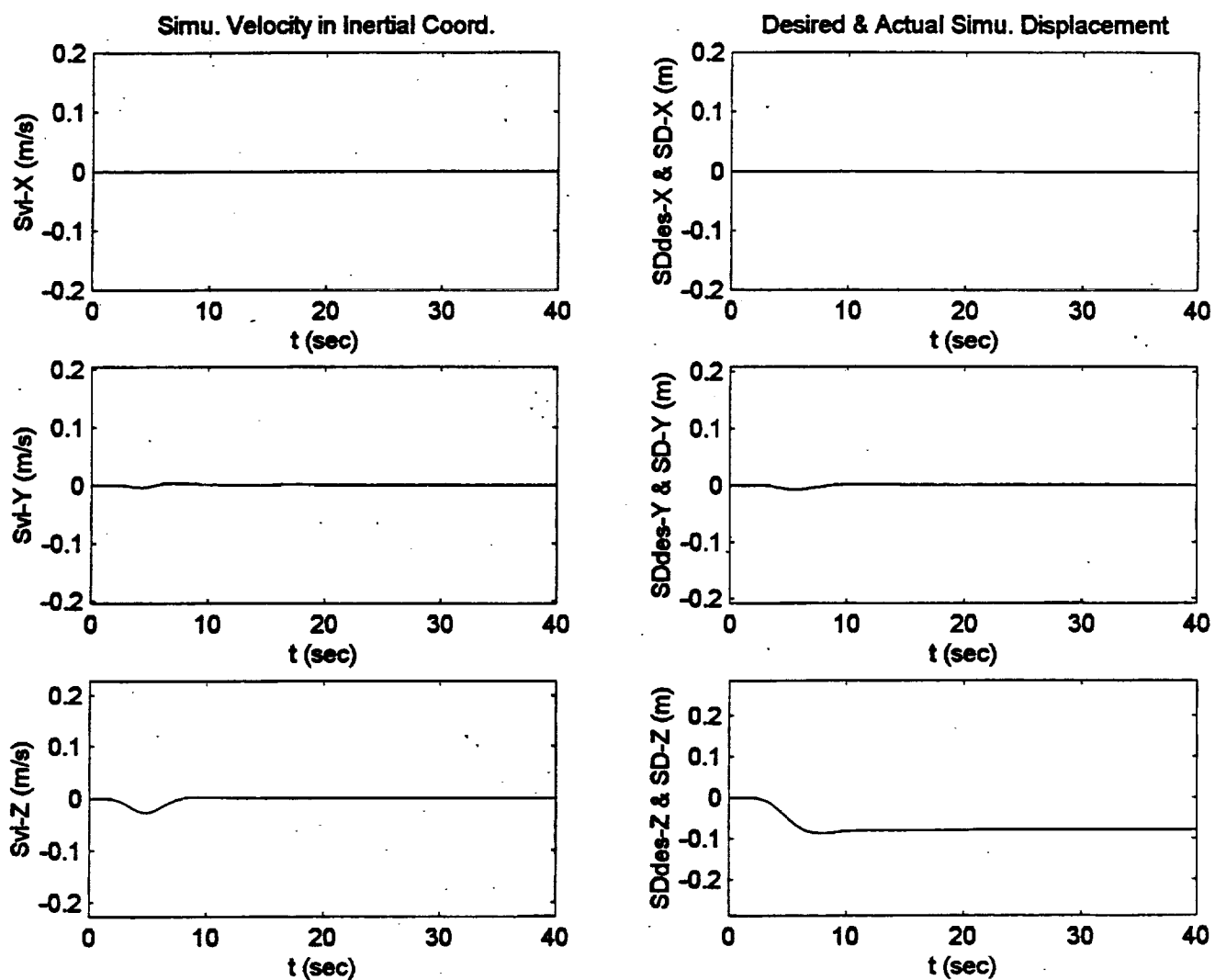


Figure C.17. NASA Adaptive Algorithm Roll Doublet Pulse Input.

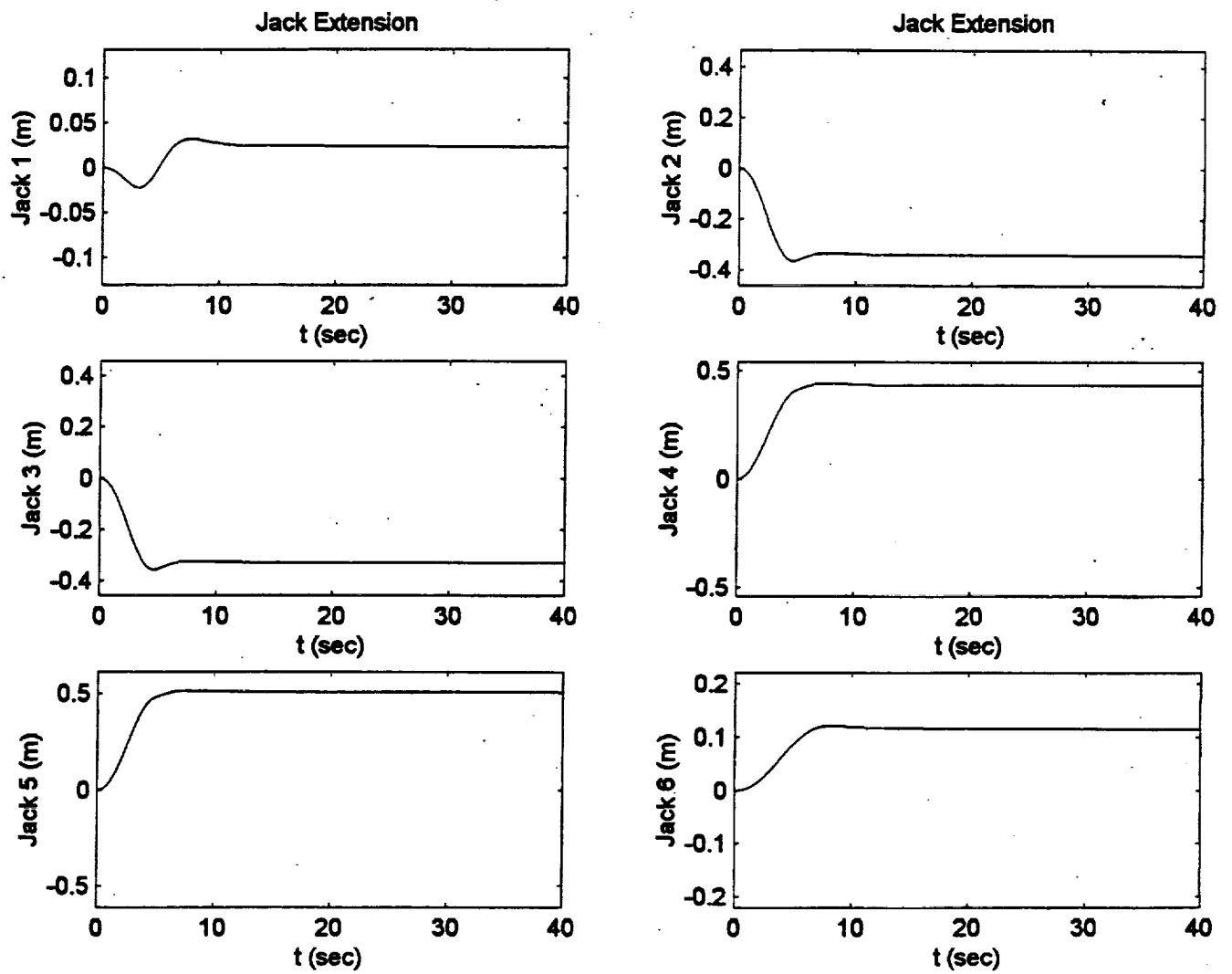


Figure C.17. NASA Adaptive Algorithm Roll Doublet Pulse Input.

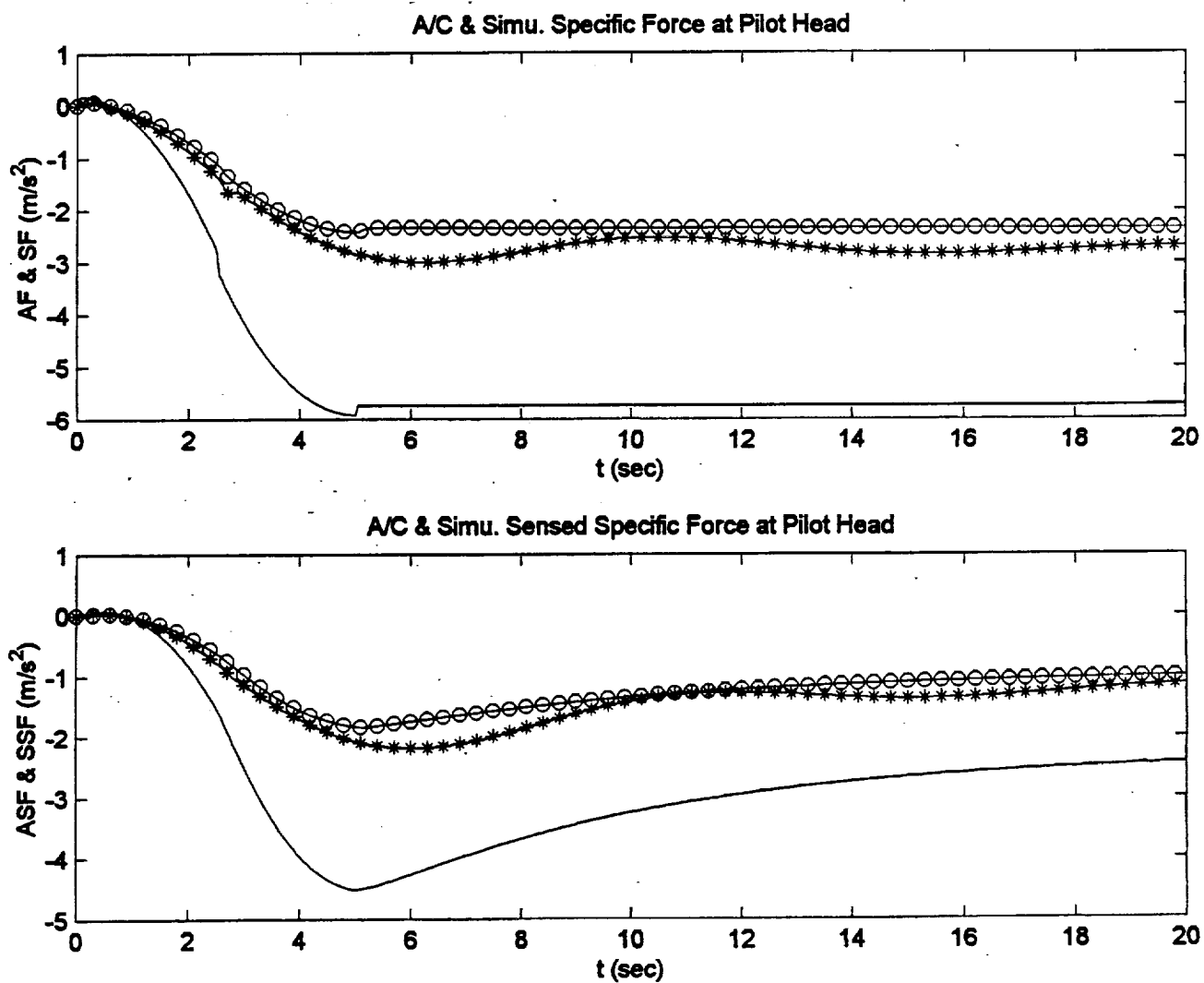


Figure C.18. Roll Doublet Pulse Input Specific Force Comparison.

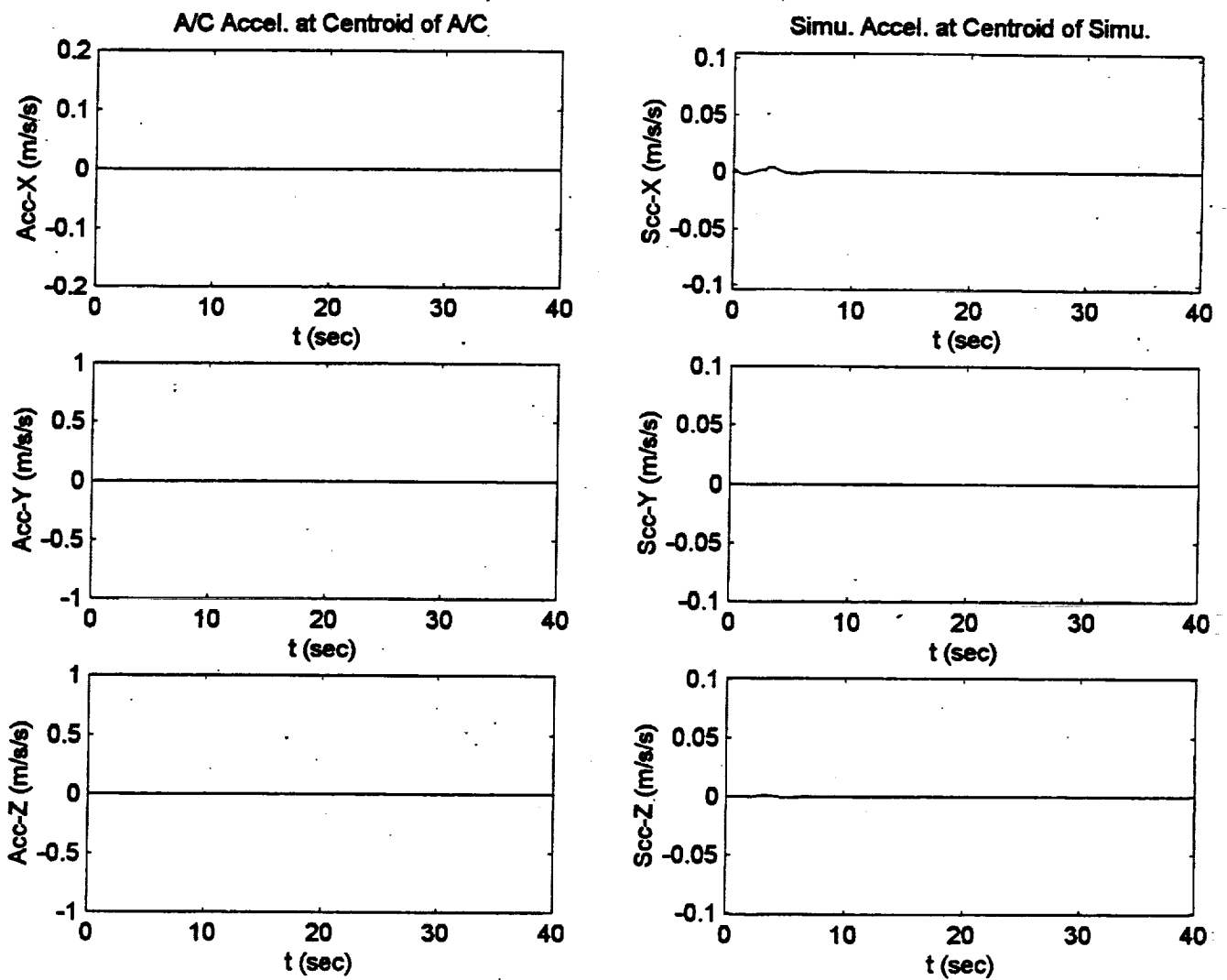


Figure C.19. Optimal Algorithm Pitch Doublet Pulse Input.

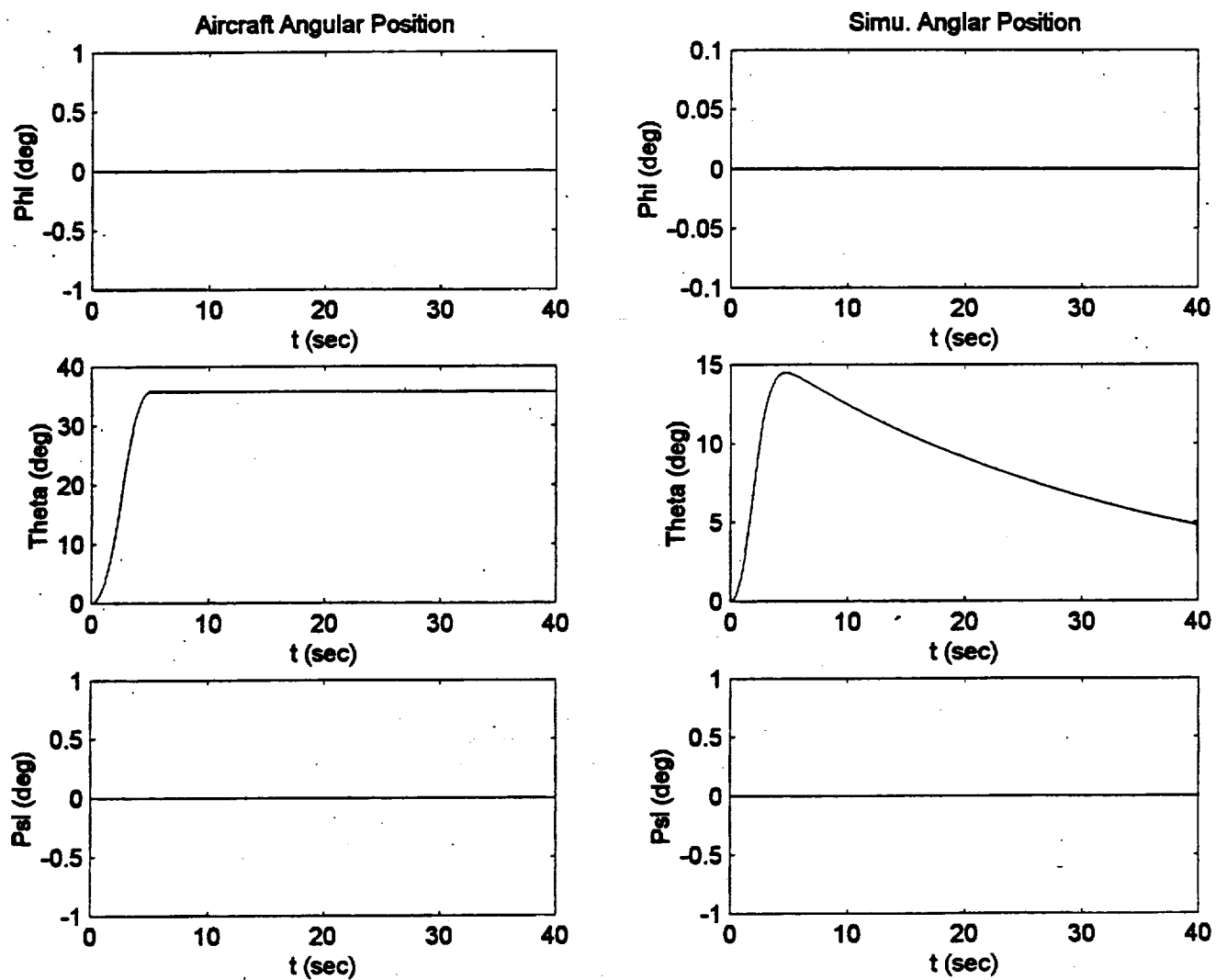


Figure C.19. Optimal Algorithm Pitch Doublet Pulse Input.

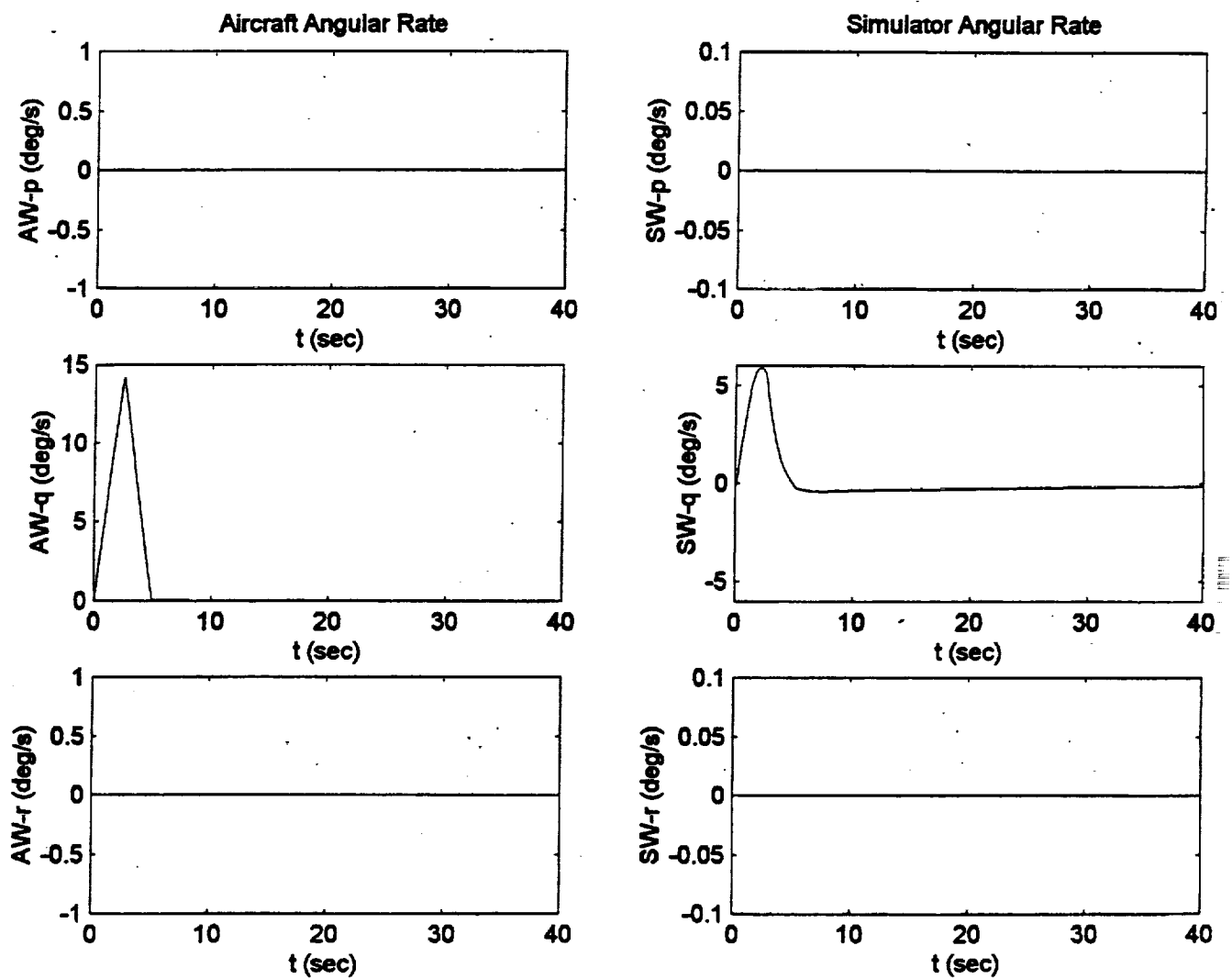


Figure C.19. Optimal Algorithm Pitch Doublet Pulse Input.

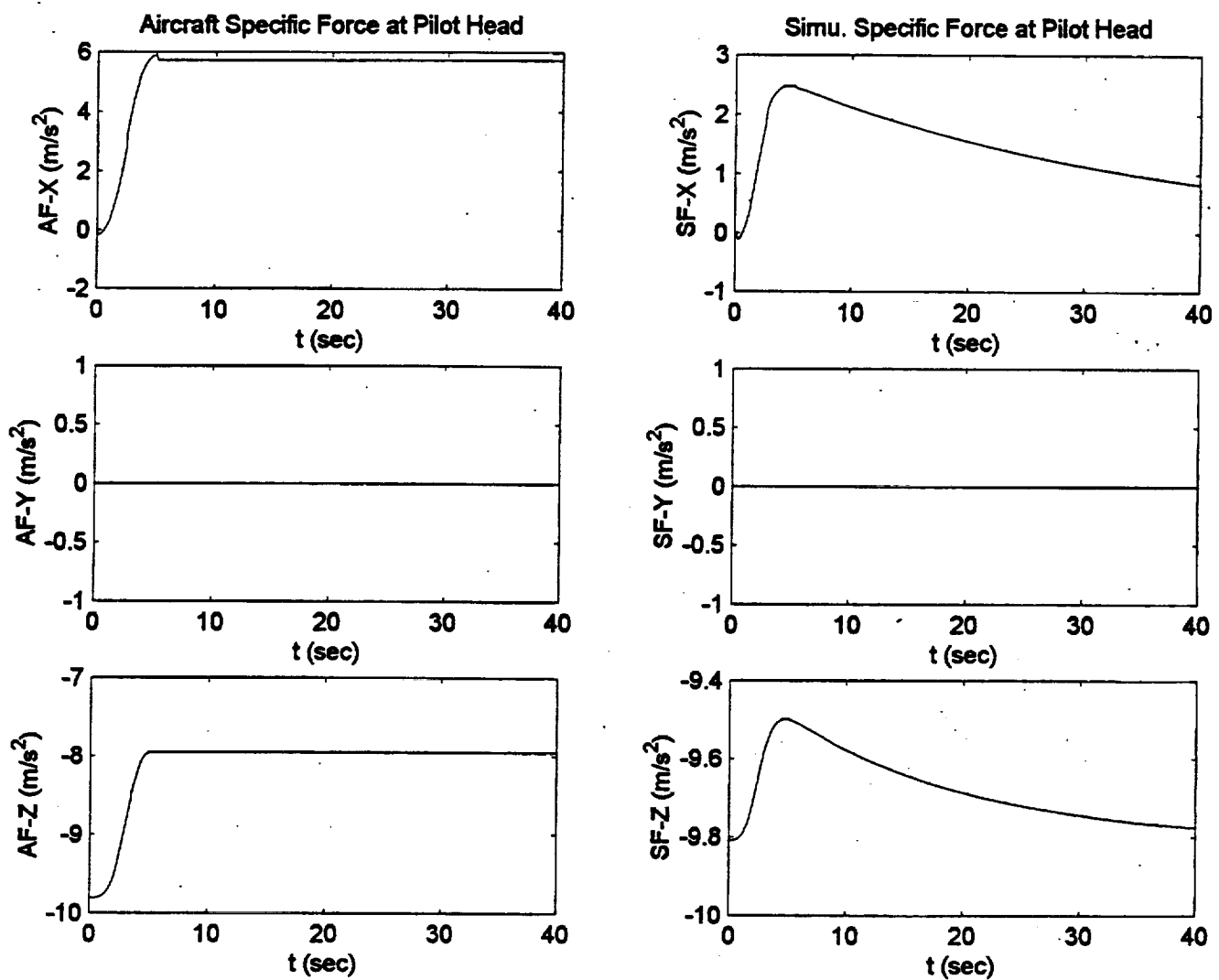


Figure C.19. Optimal Algorithm Pitch Doublet Pulse Input.

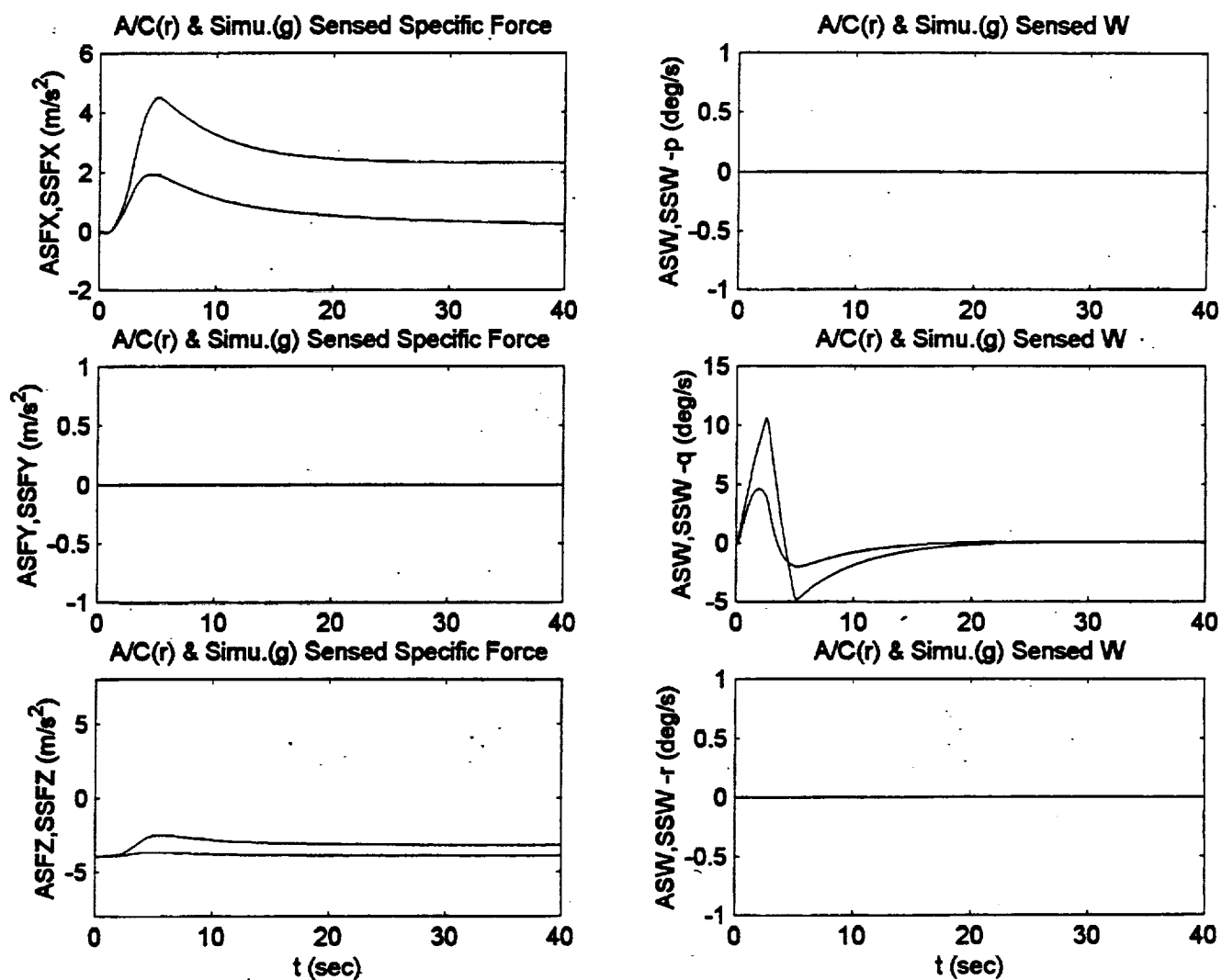


Figure C.19. Optimal Algorithm Pitch Doublet Pulse Input.

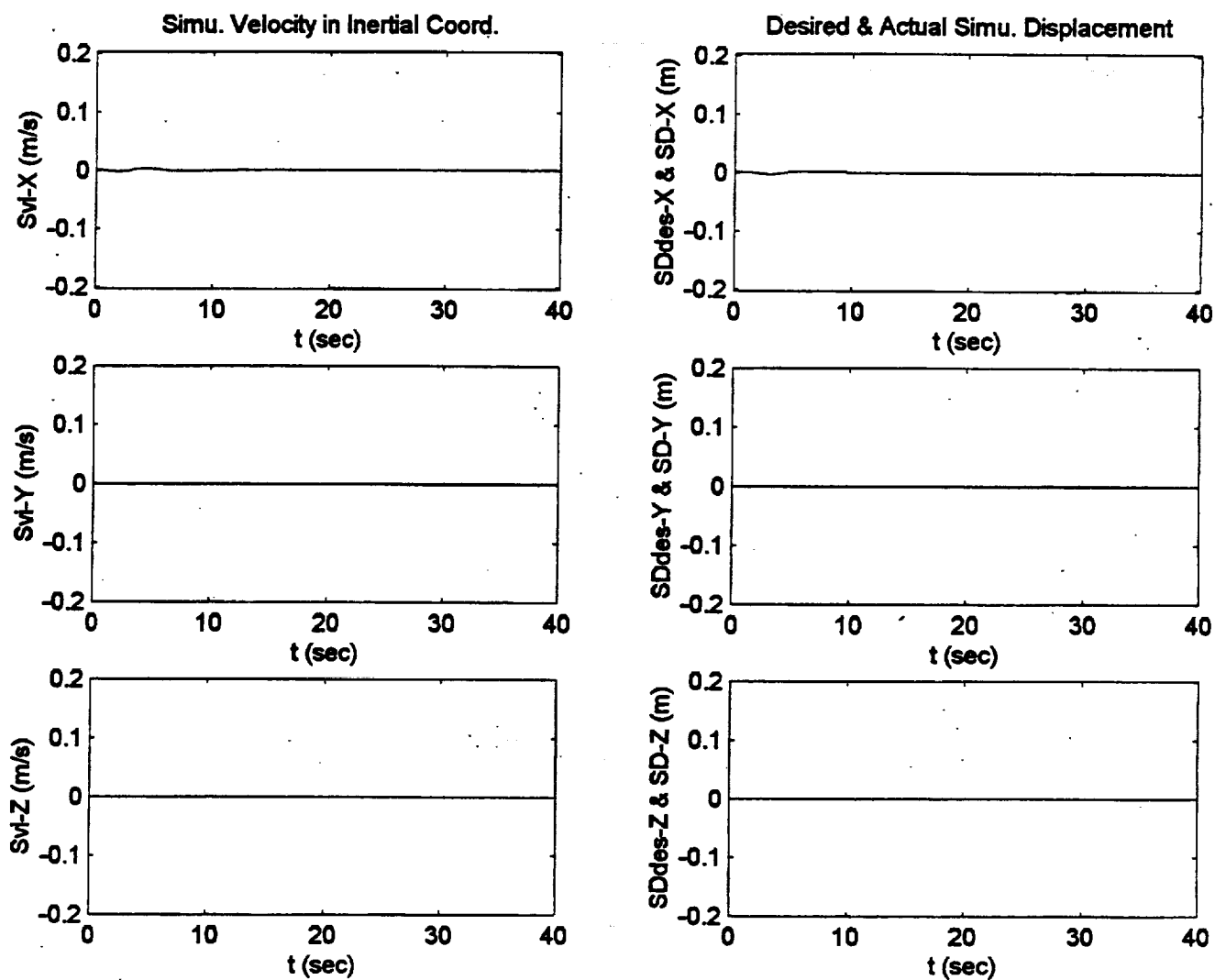


Figure C.19. Optimal Algorithm Pitch Doublet Pulse Input.

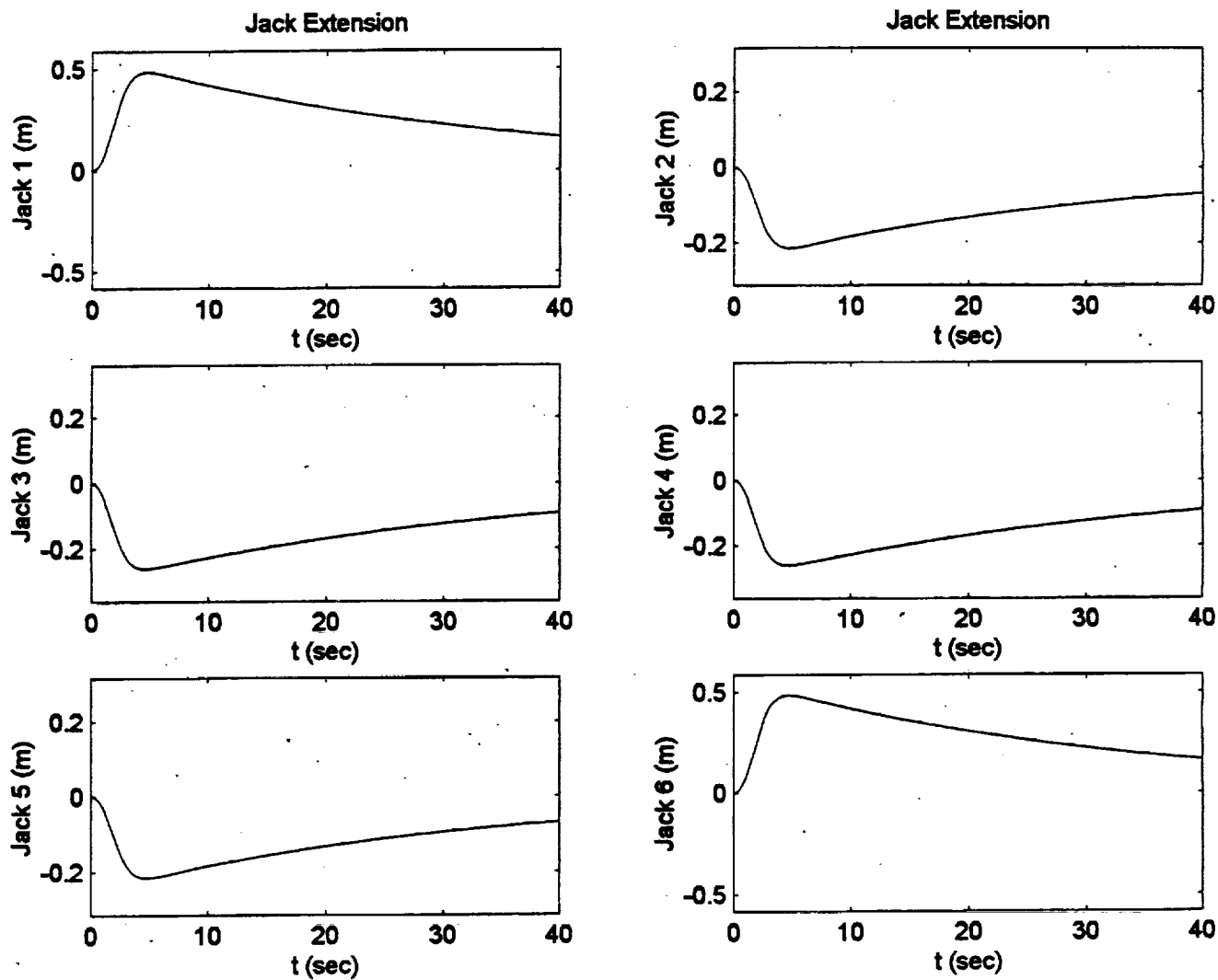


Figure C.19. Optimal Algorithm Pitch Doublet Pulse Input.

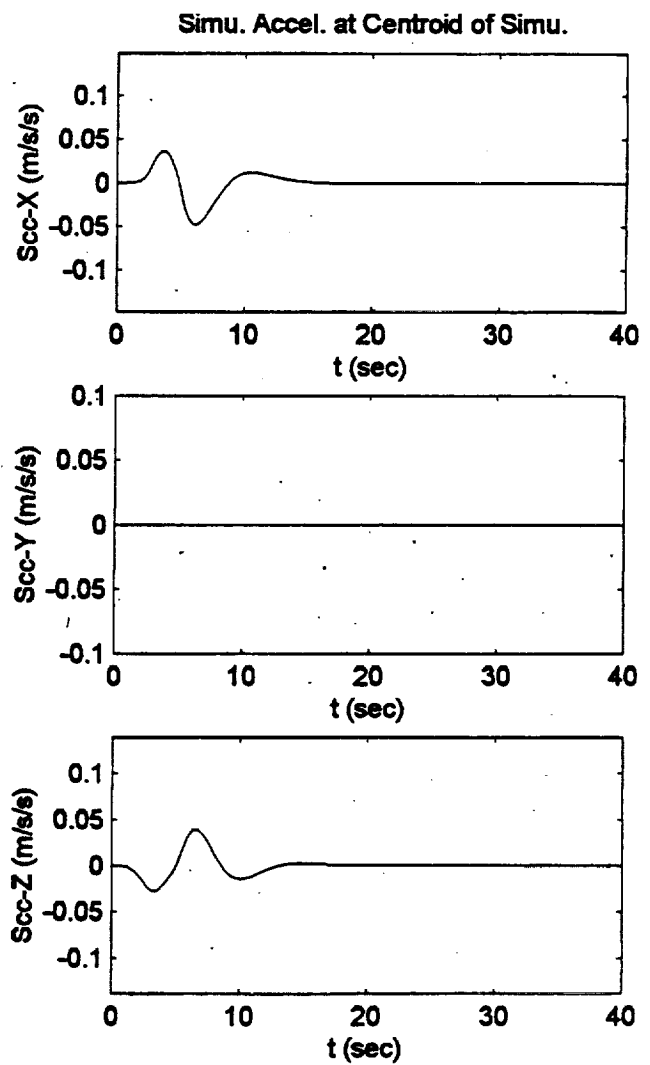
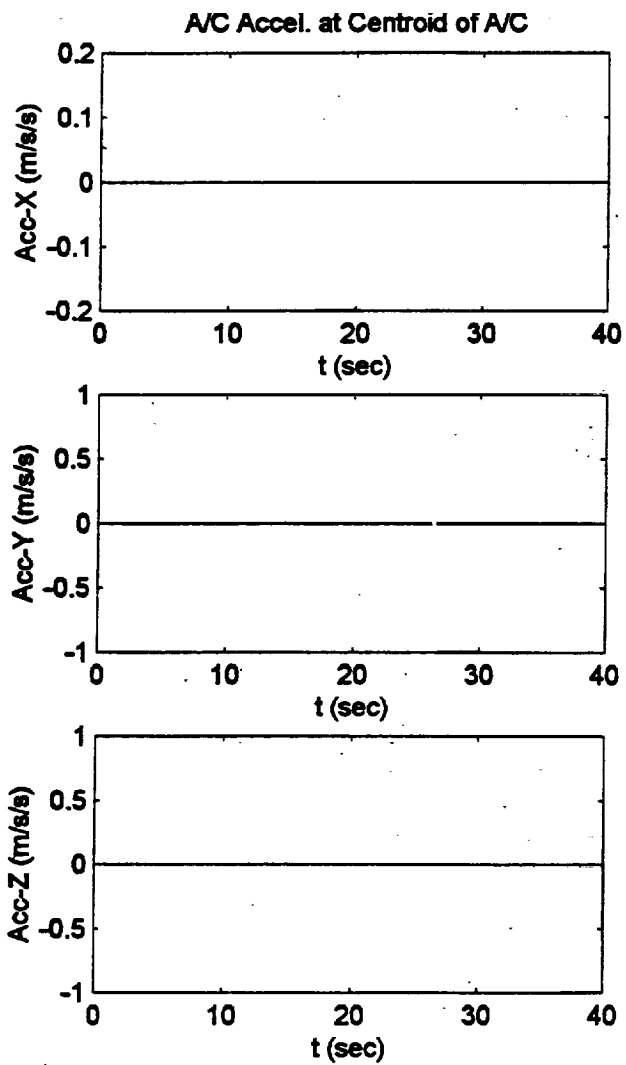


Figure C.20. NASA Adaptive Algorithm Pitch Doublet Pulse Input.

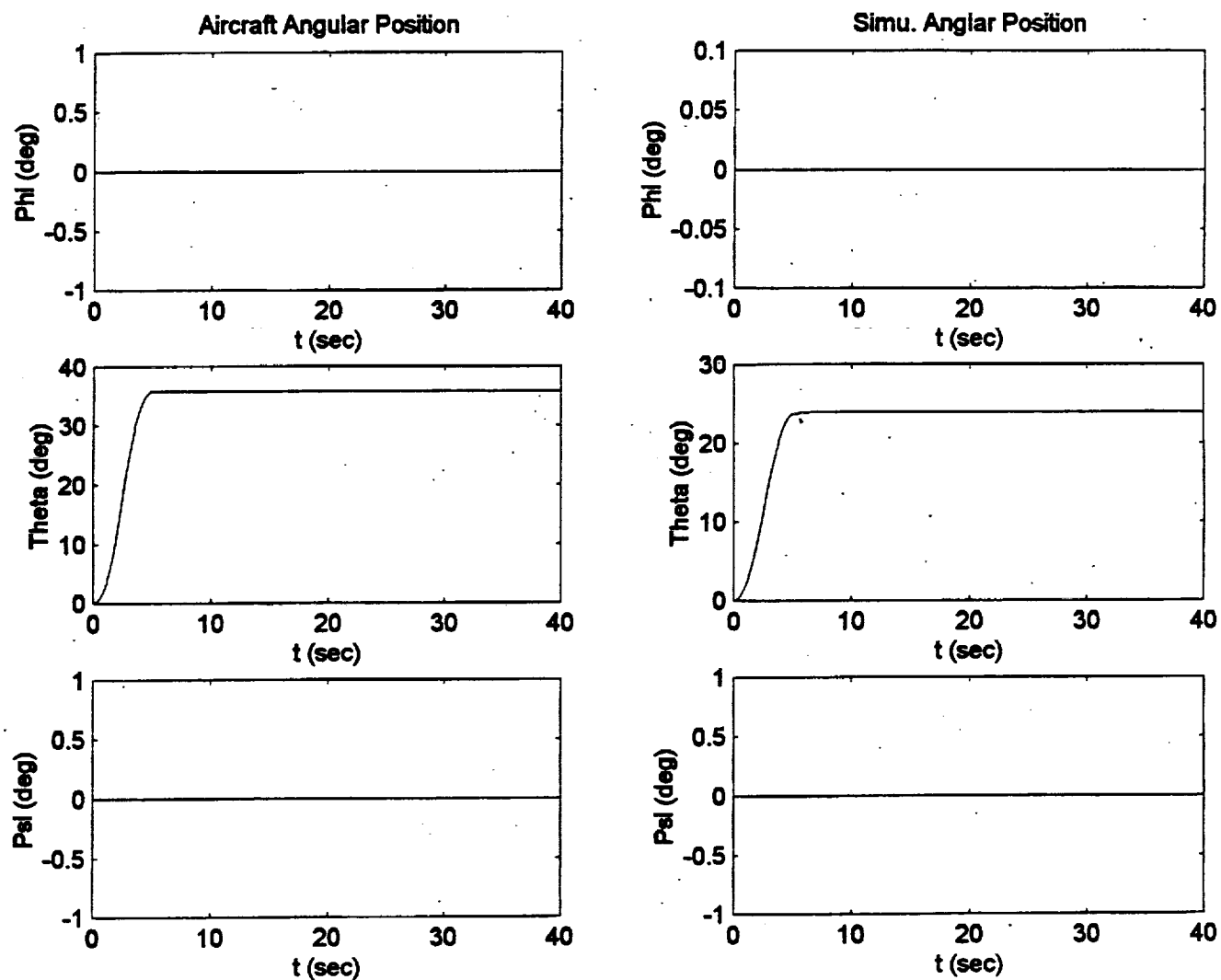


Figure C.20. NASA Adaptive Algorithm Pitch Doublet Pulse Input.

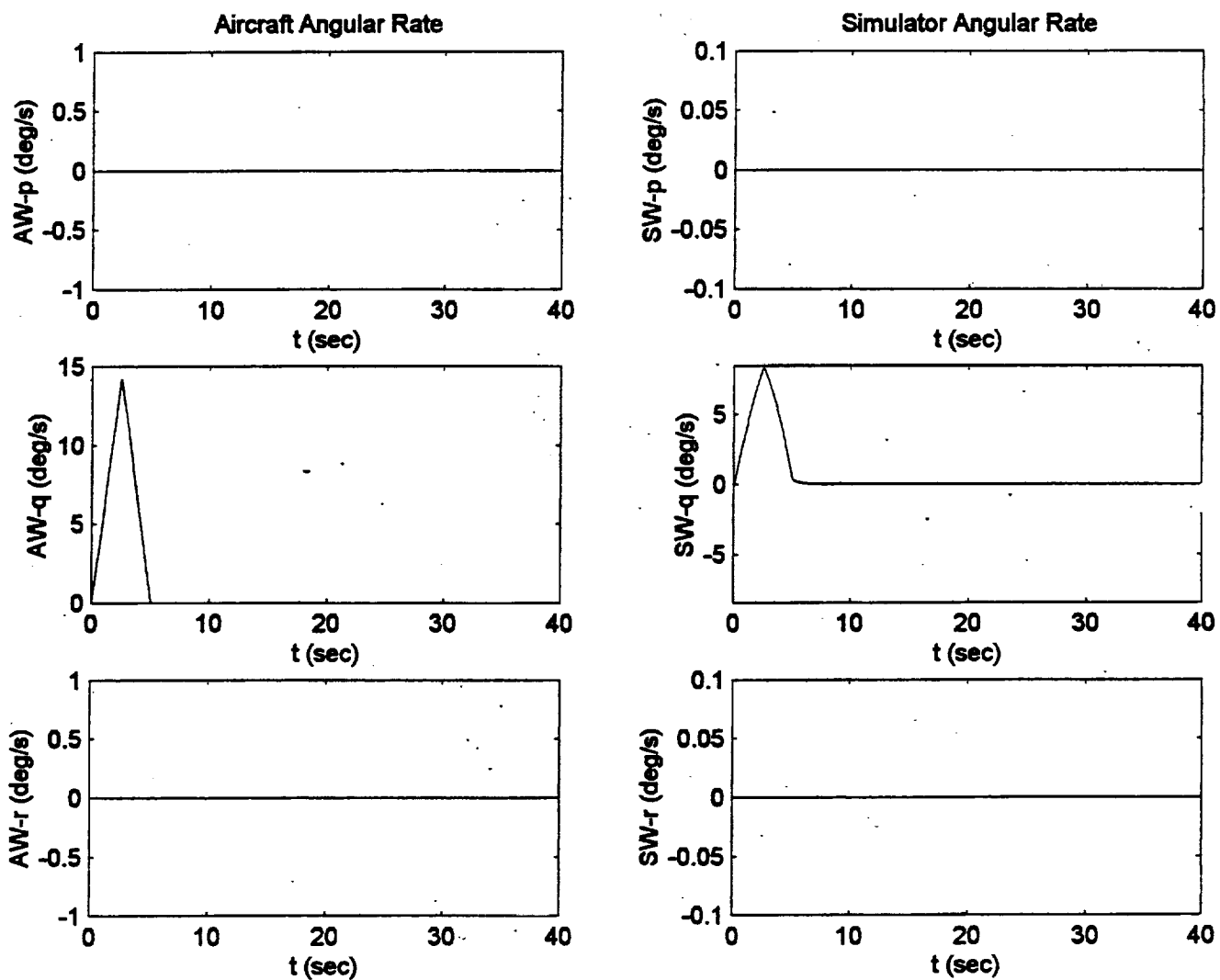


Figure C.20. NASA Adaptive Algorithm Pitch Doublet Pulse Input.

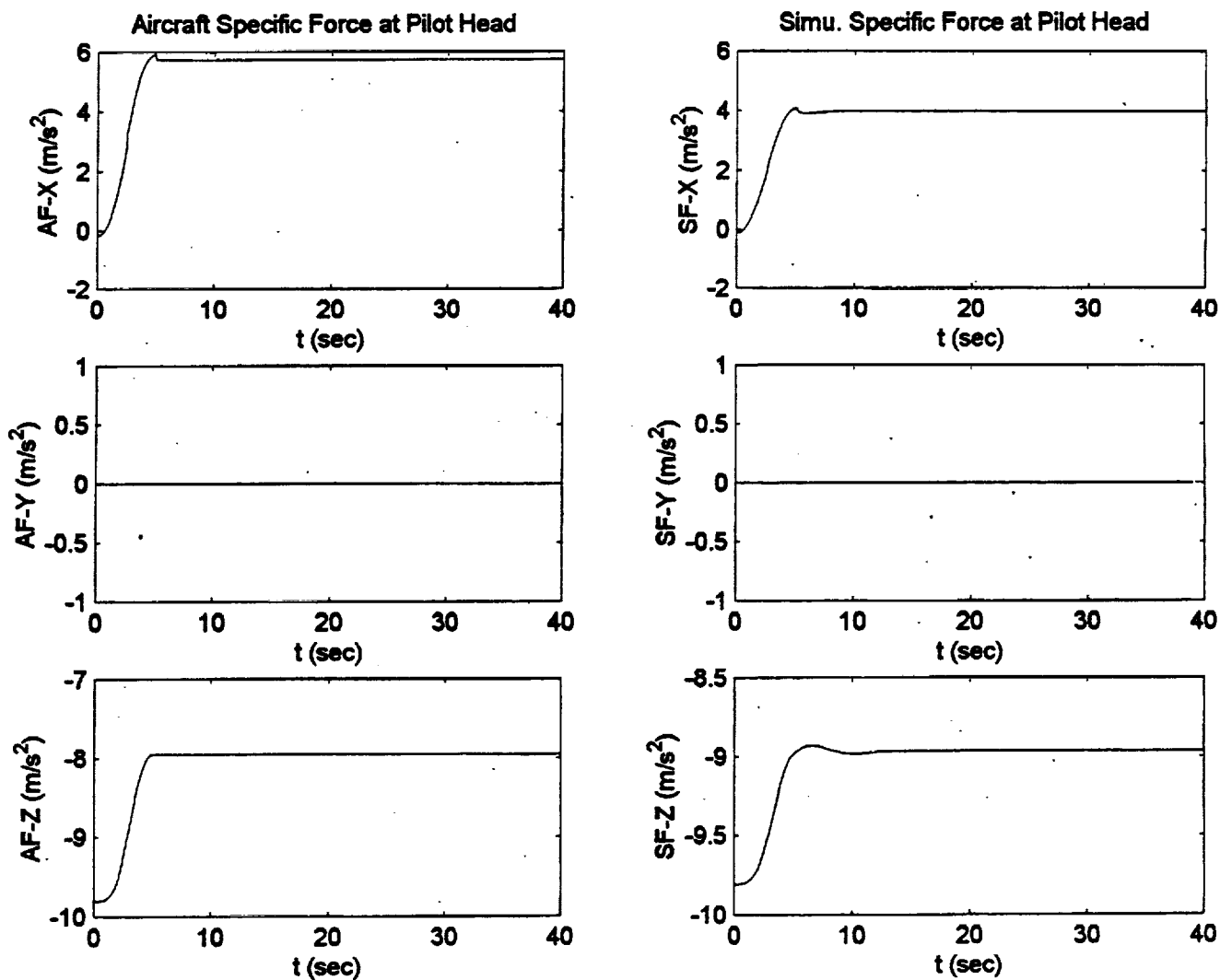


Figure C.20. NASA Adaptive Algorithm Pitch Doublet Pulse Input.

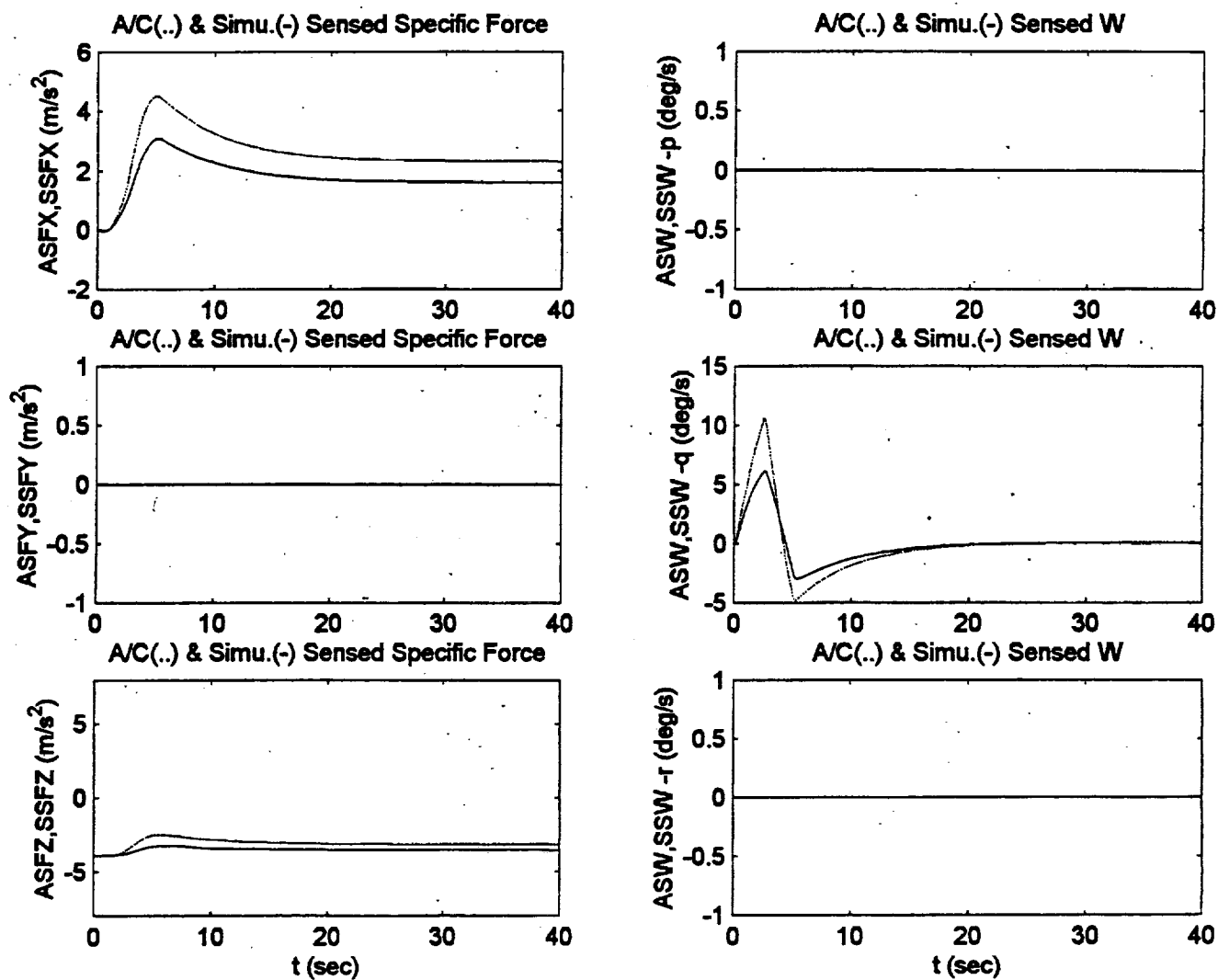


Figure C.20. NASA Adaptive Algorithm Pitch Doublet Pulse Input.

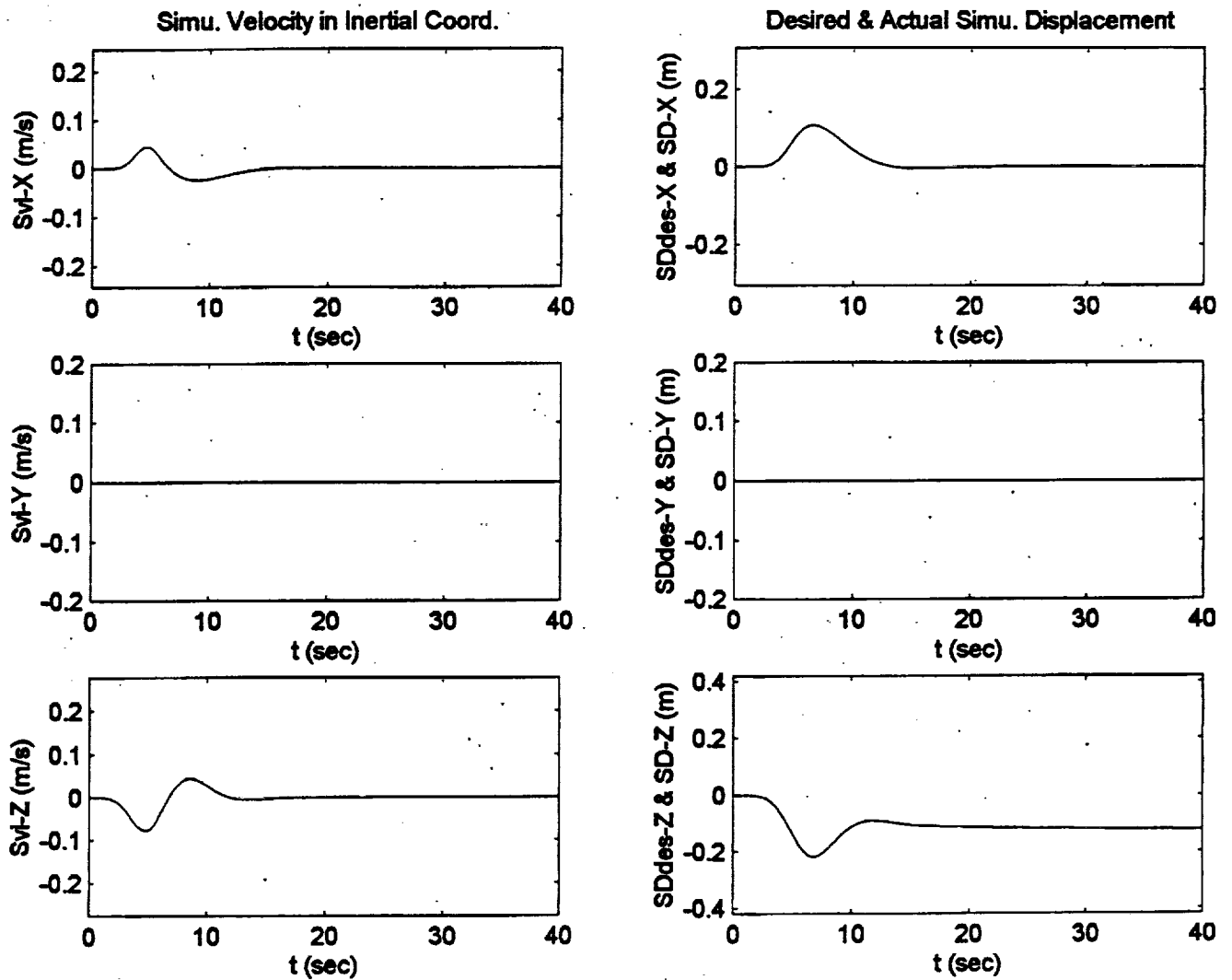


Figure C.20. NASA Adaptive Algorithm Pitch Doublet Pulse Input.

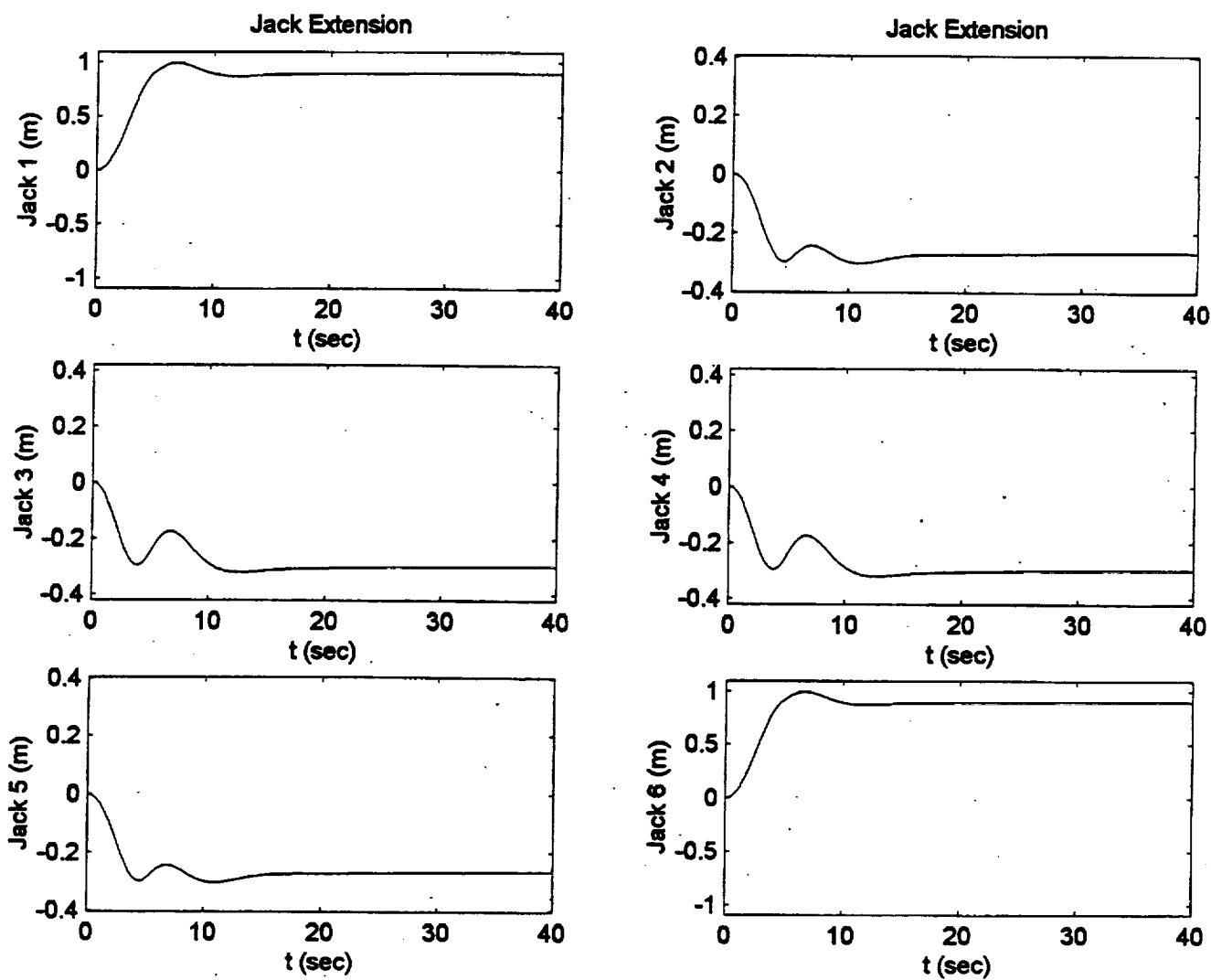


Figure C.20. NASA Adaptive Algorithm Pitch Doublet Pulse Input.

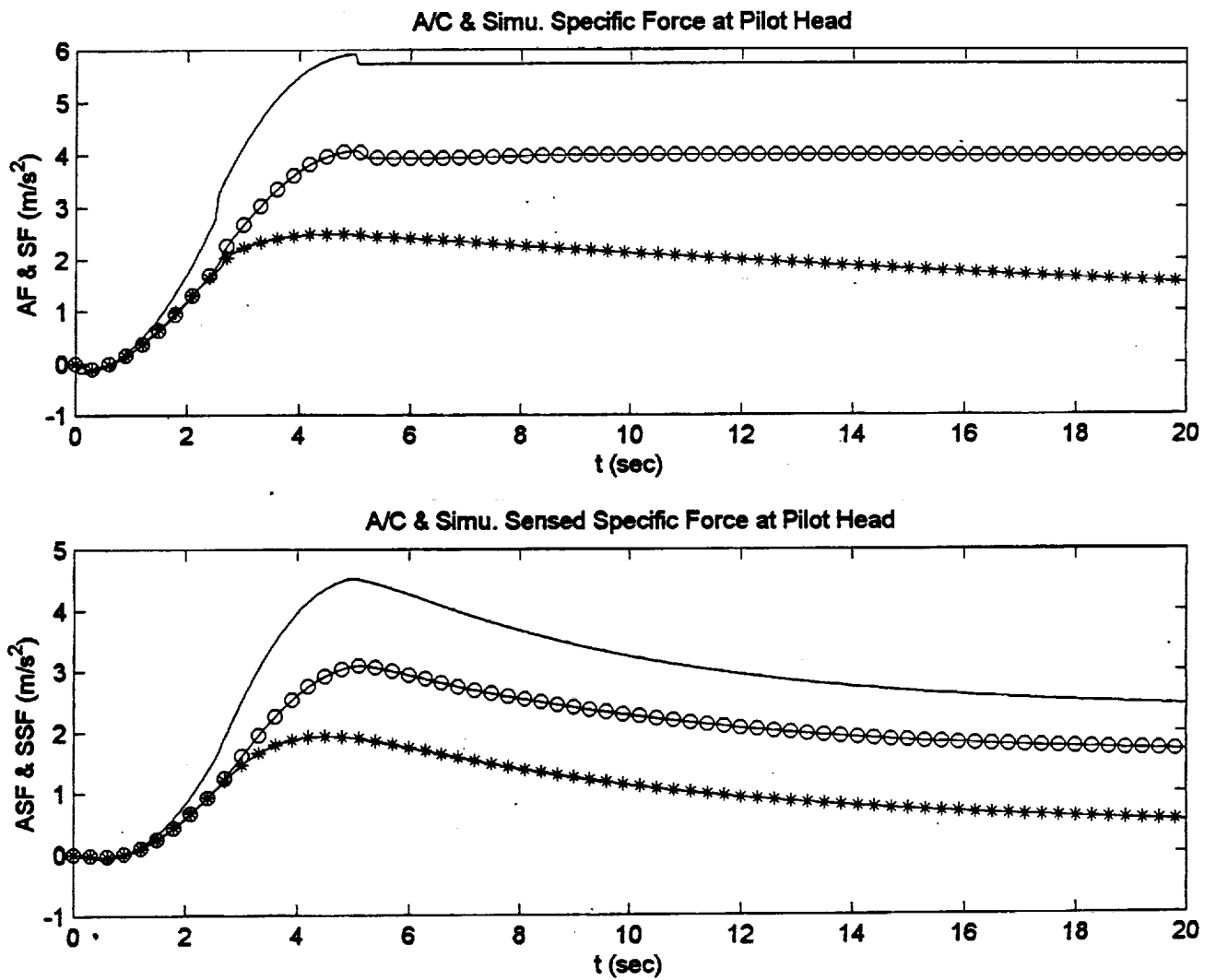


Figure C21. Pitch Doublet Pulse Input Specific Force Comparison.

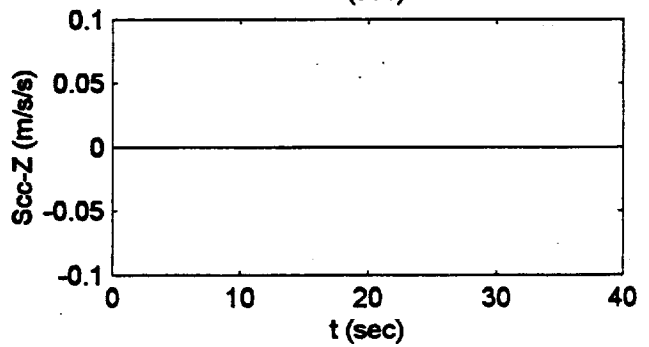
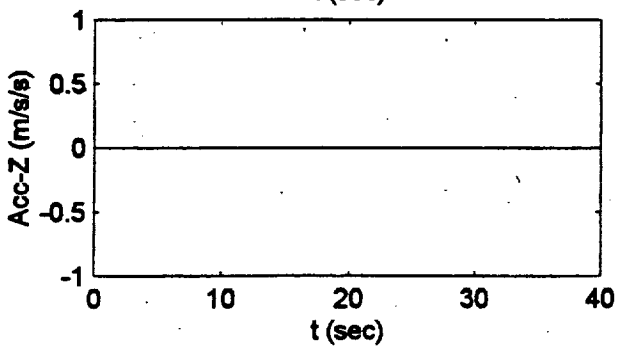
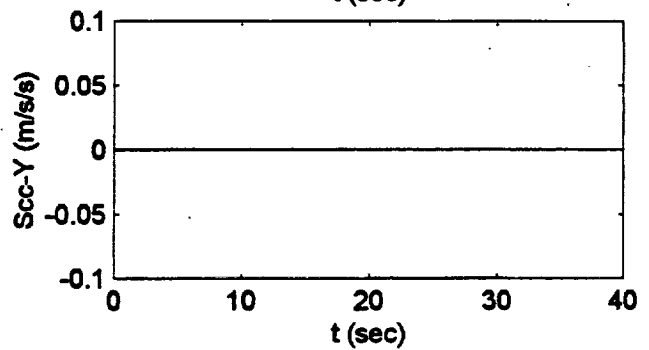
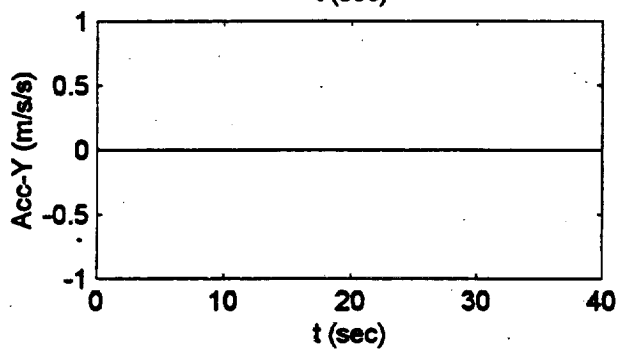
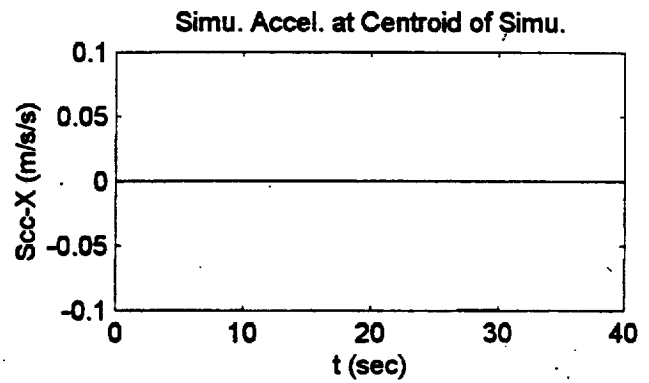
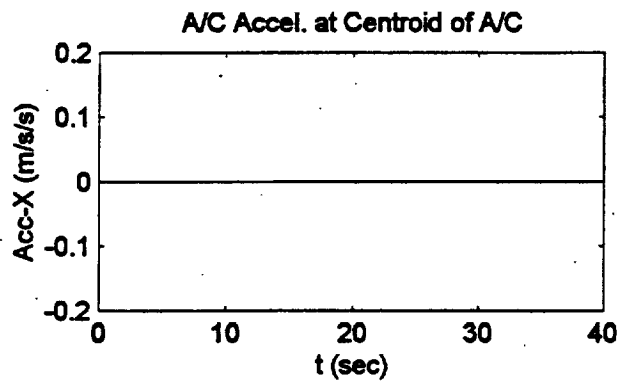


Figure C22. Optimal Algorithm Yaw Doublet Pulse Input.

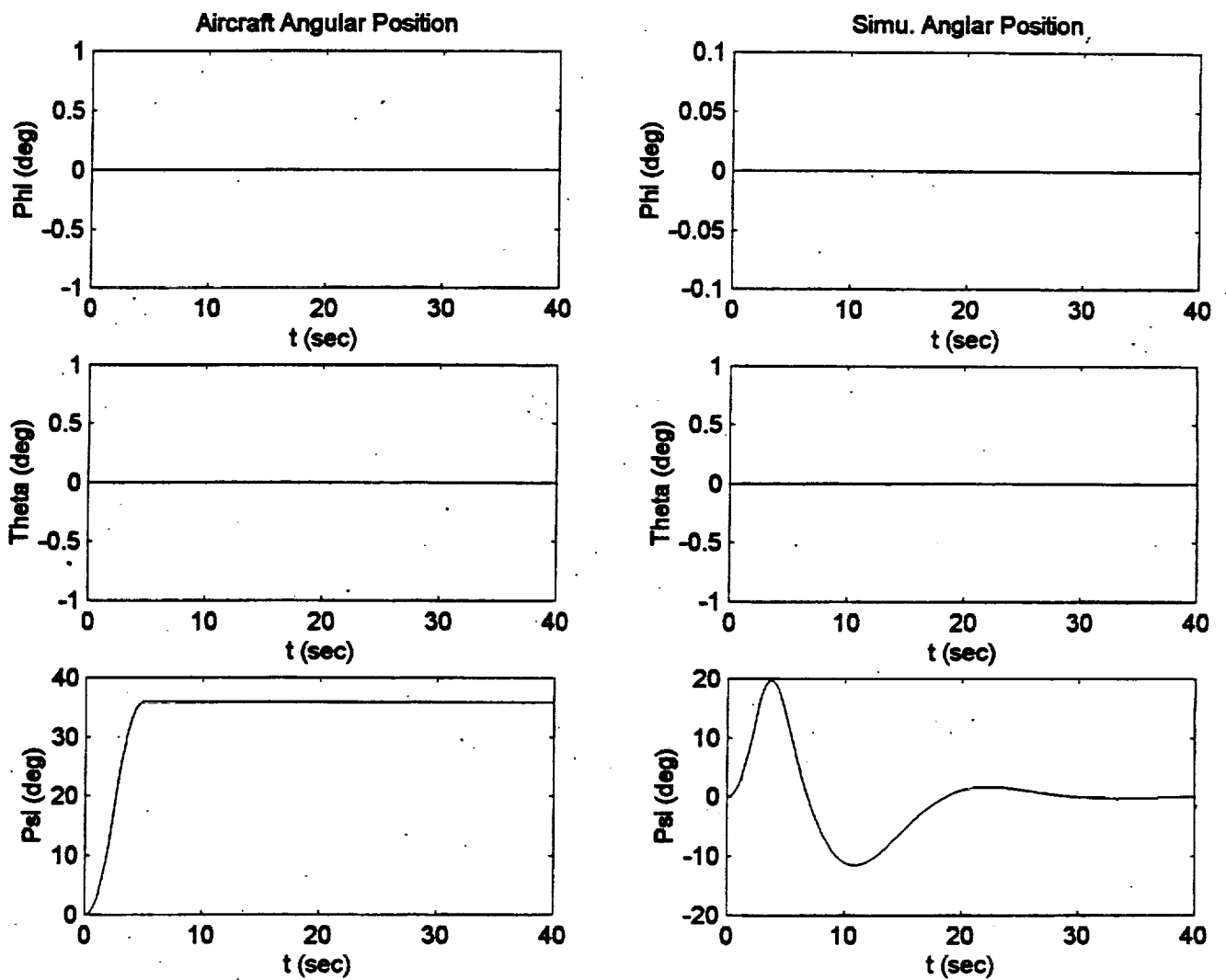


Figure C22. Optimal Algorithm Yaw Doublet Pulse Input.

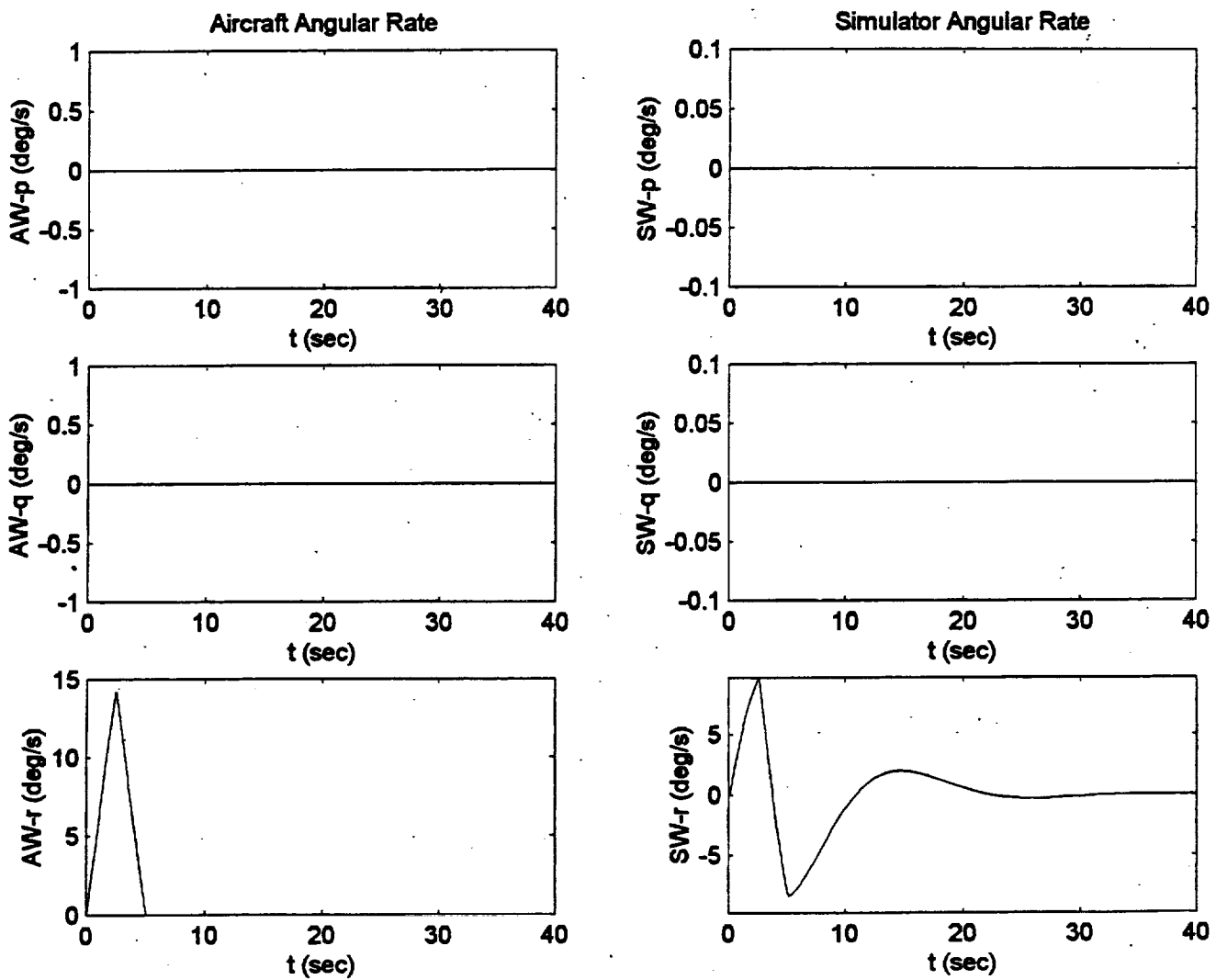


Figure C22. Optimal Algorithm Yaw Doublet Pulse Input.

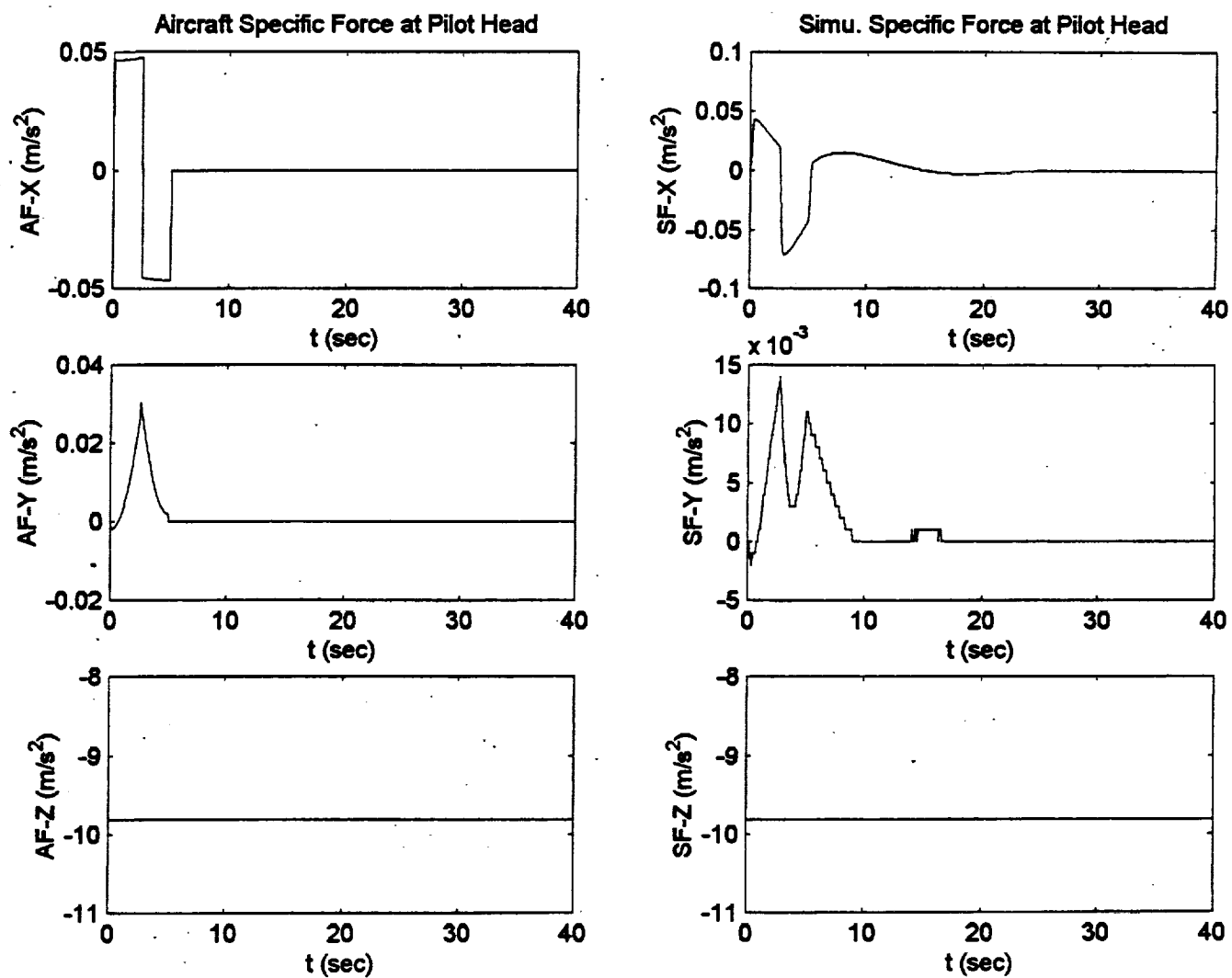


Figure C22. Optimal Algorithm Yaw Doublet Pulse Input.

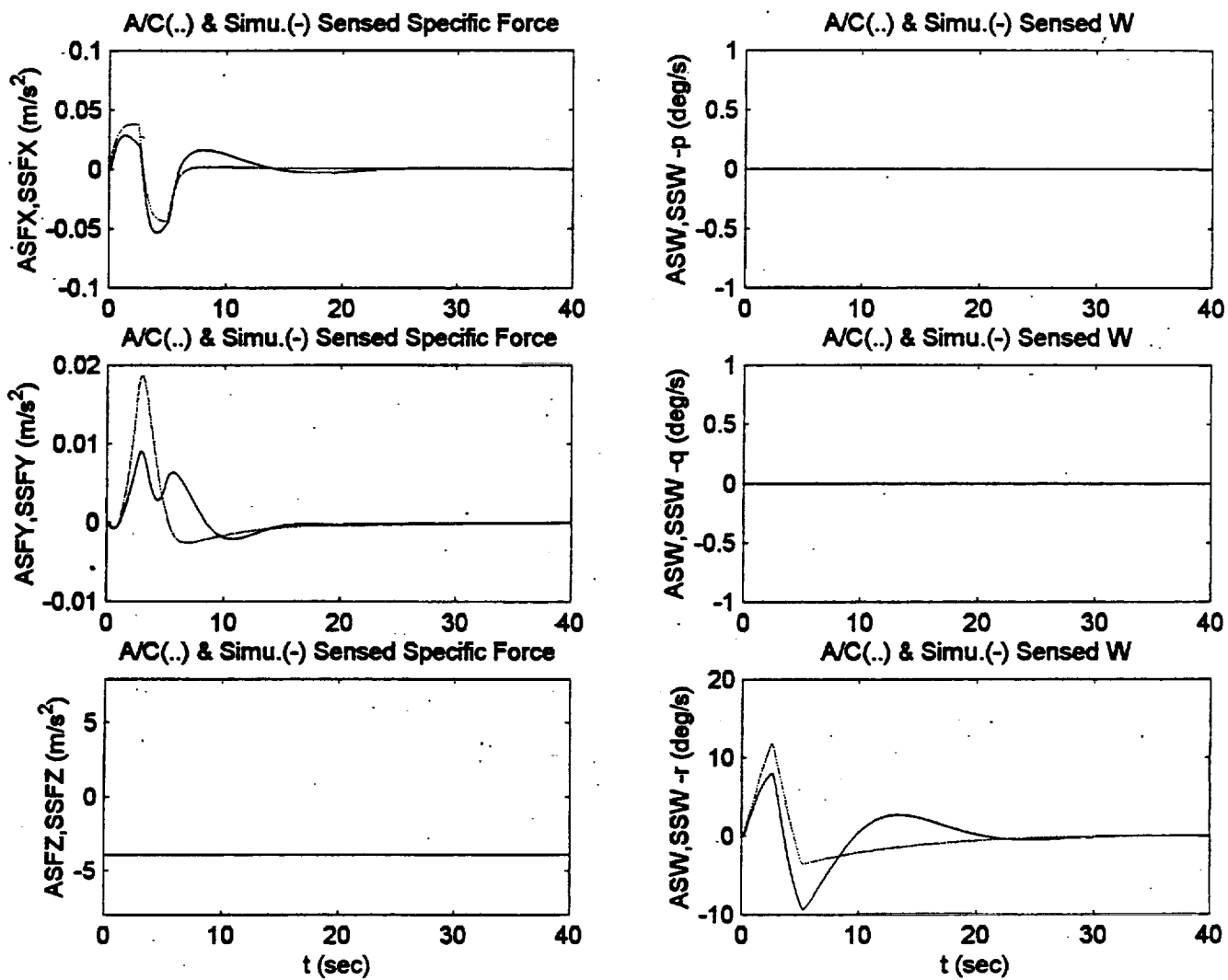


Figure C22. Optimal Algorithm Yaw Doublet Pulse Input.

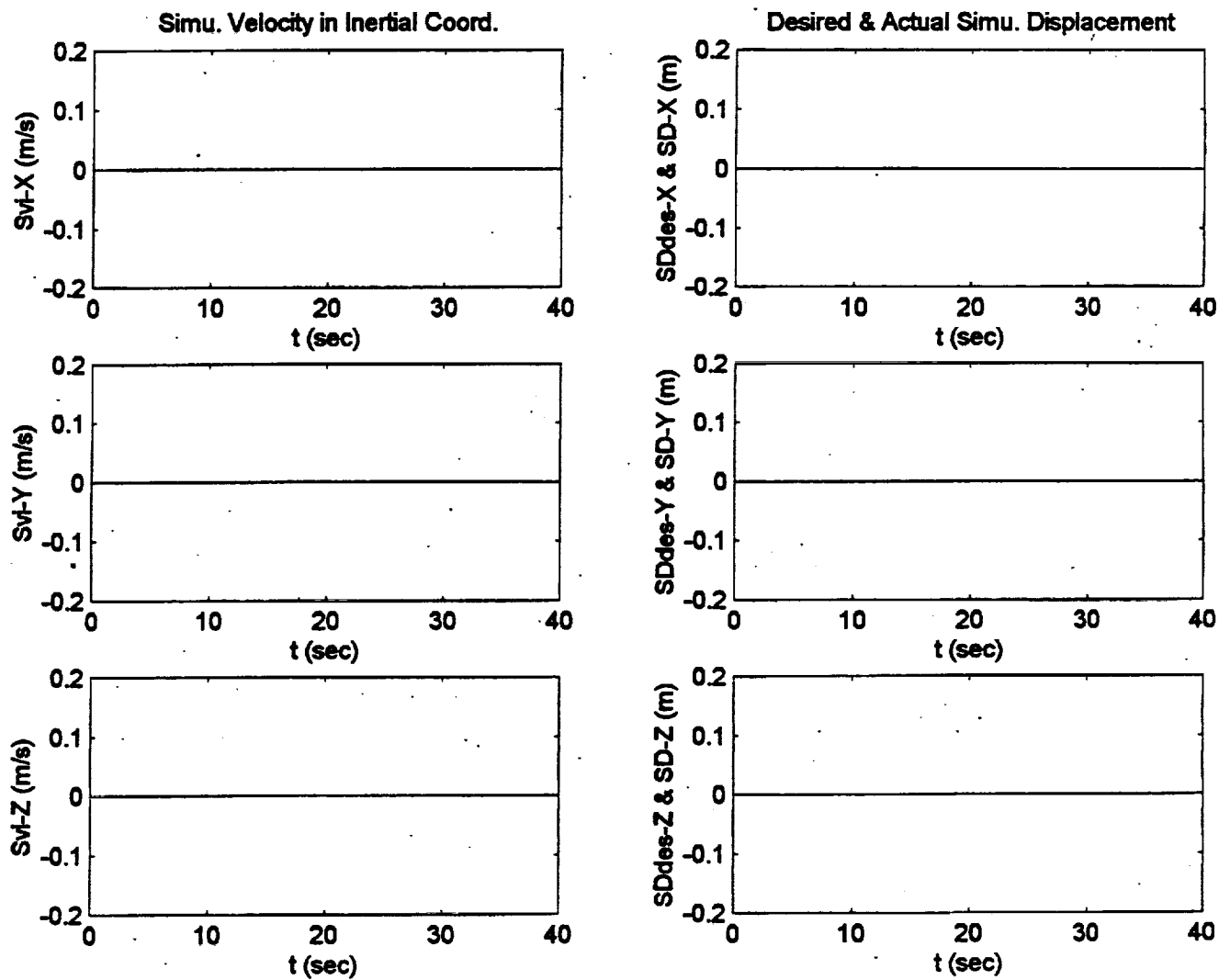


Figure C22. Optimal Algorithm Yaw Doublet Pulse Input.

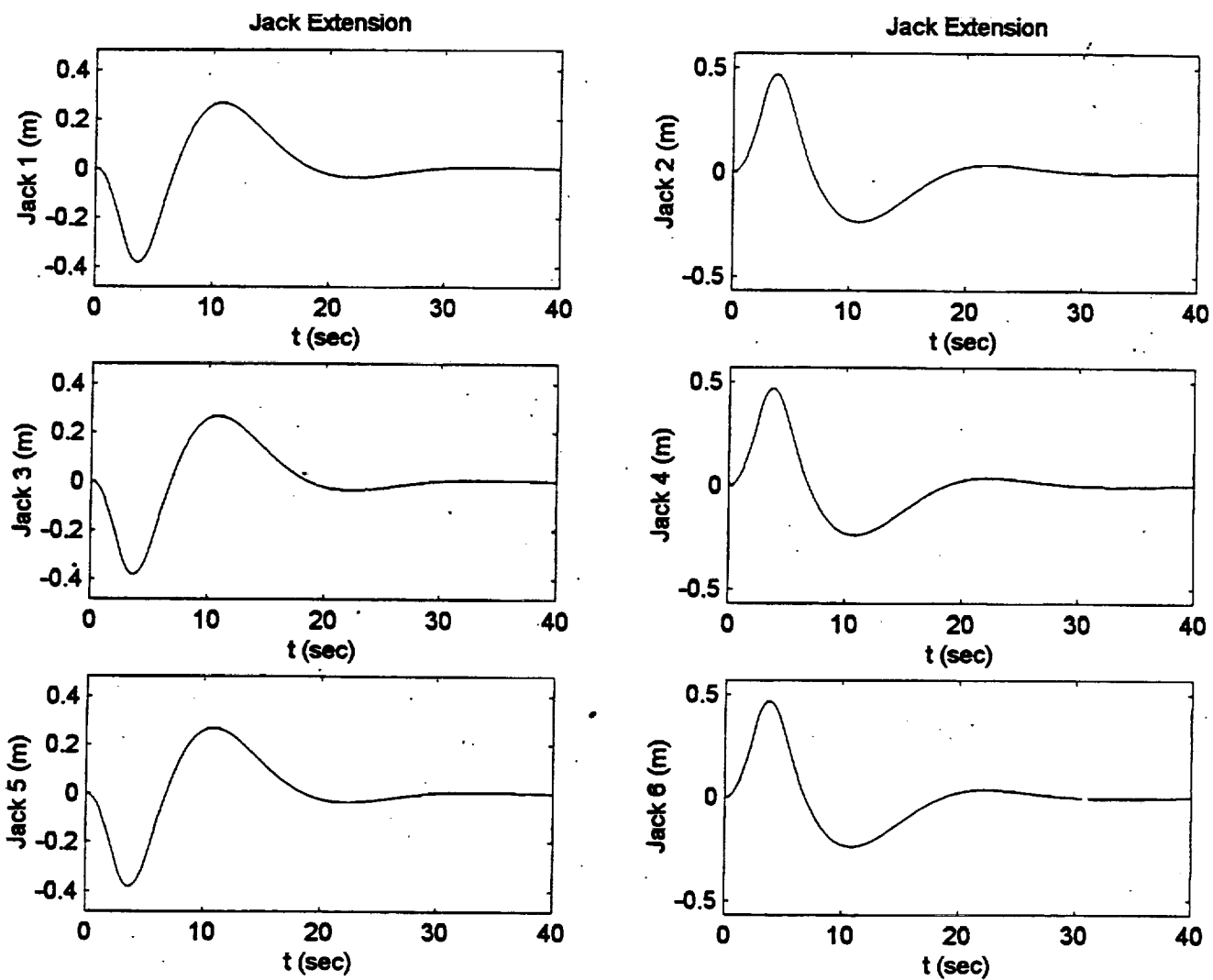


Figure C22. Optimal Algorithm Yaw Doublet Pulse Input.

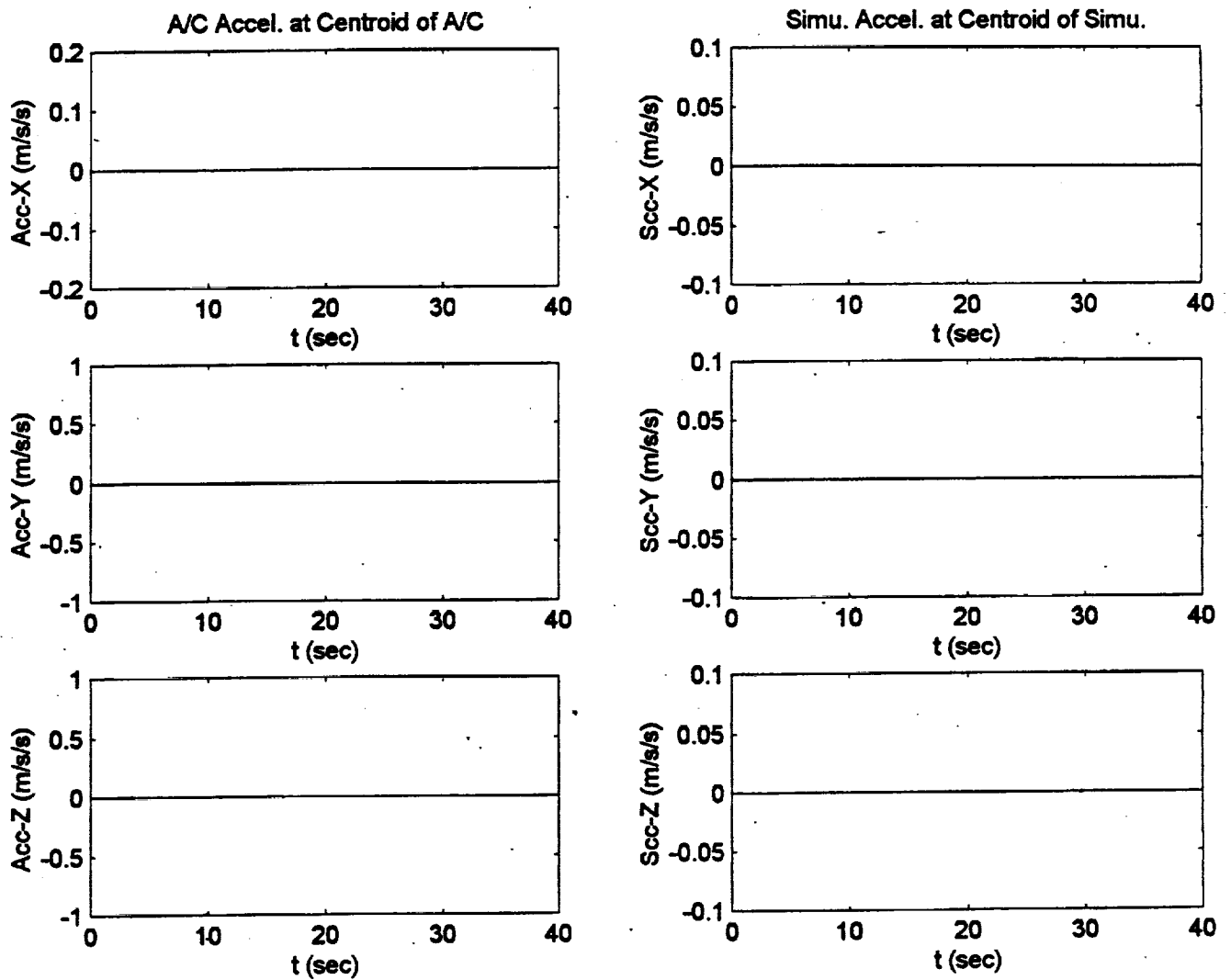


Figure C.23. NASA Adaptive Algorithm Yaw Doublet Pulse Input.

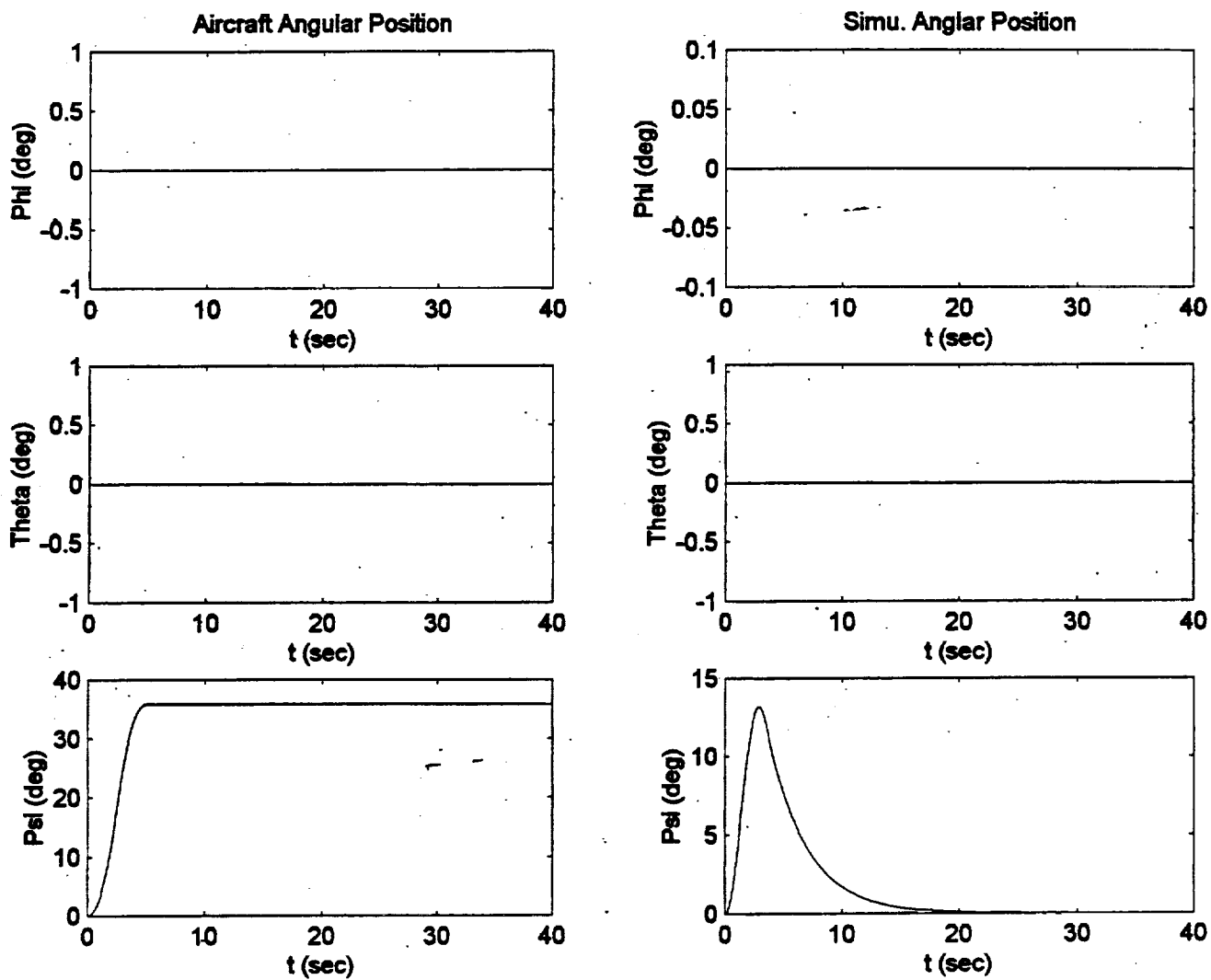


Figure C.23. NASA Adaptive Algorithm Yaw Doublet Pulse Input.

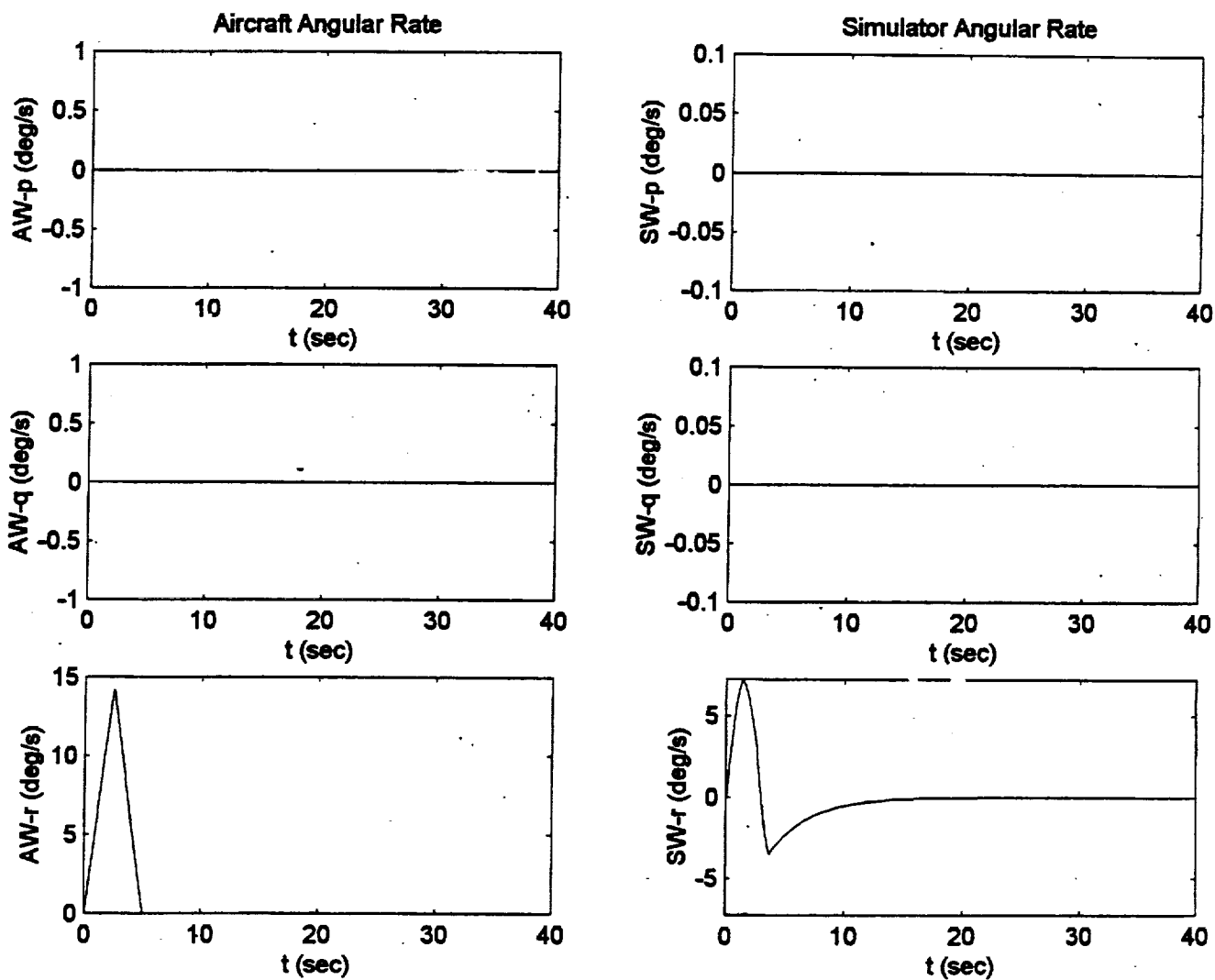


Figure C.23. NASA Adaptive Algorithm Yaw Doublet Pulse Input.

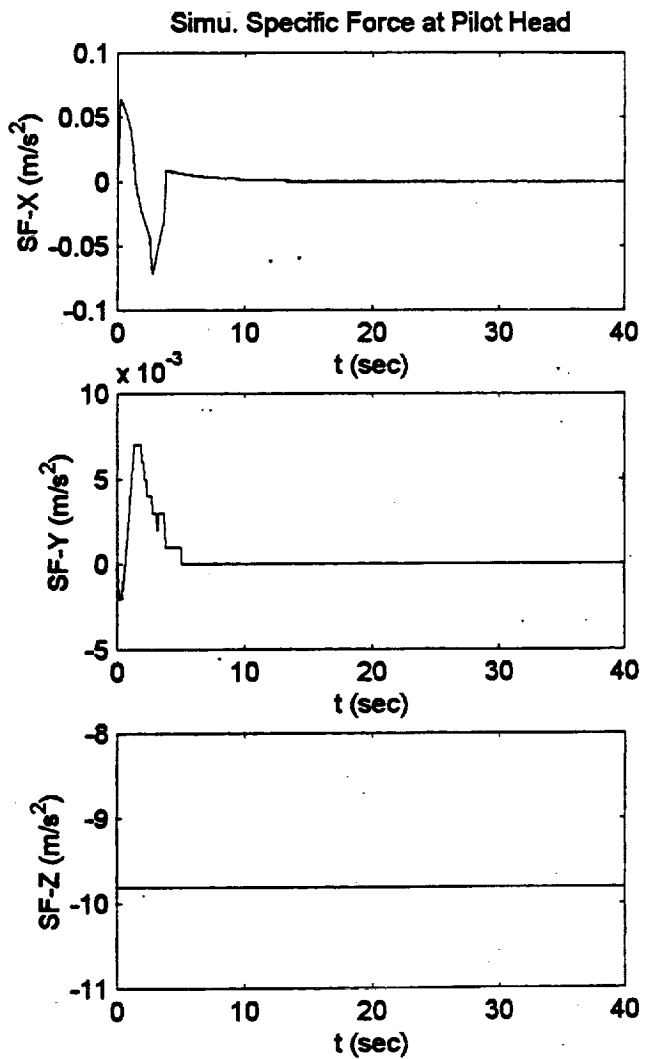
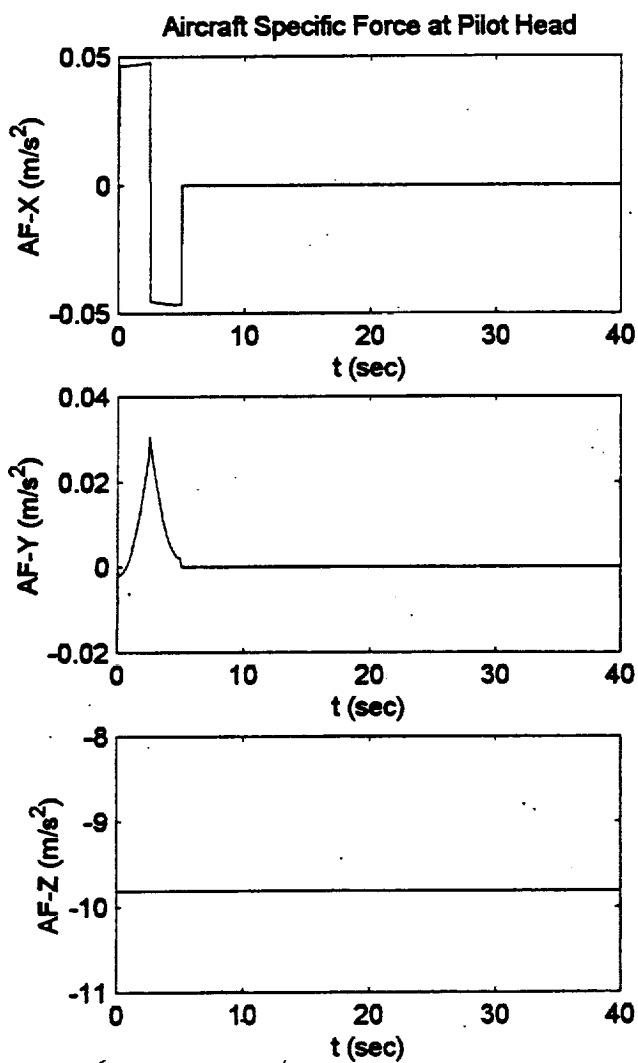


Figure C.23. NASA Adaptive Algorithm Yaw Doublet Pulse Input.

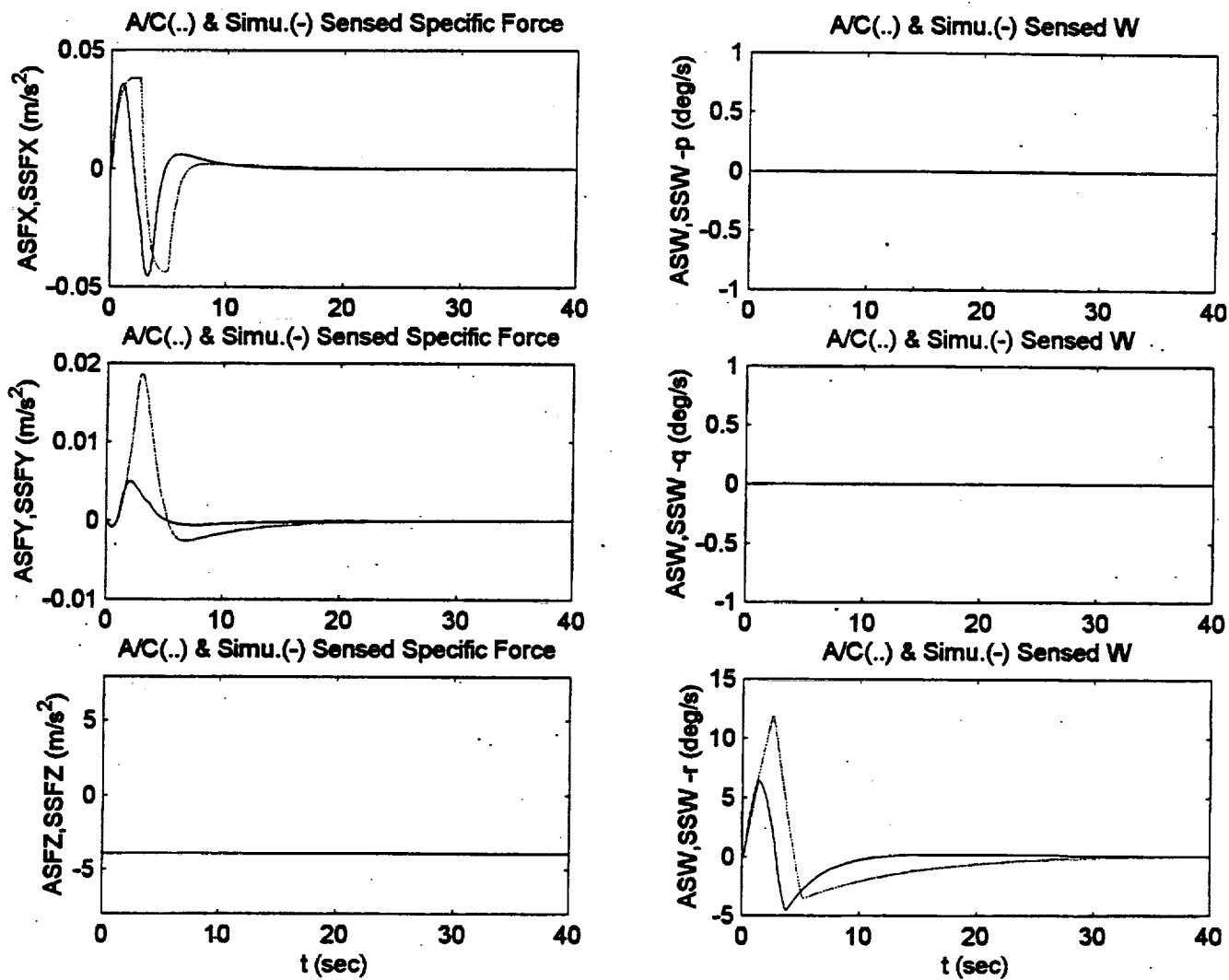


Figure C.23. NASA Adaptive Algorithm Yaw Doublet Pulse Input.

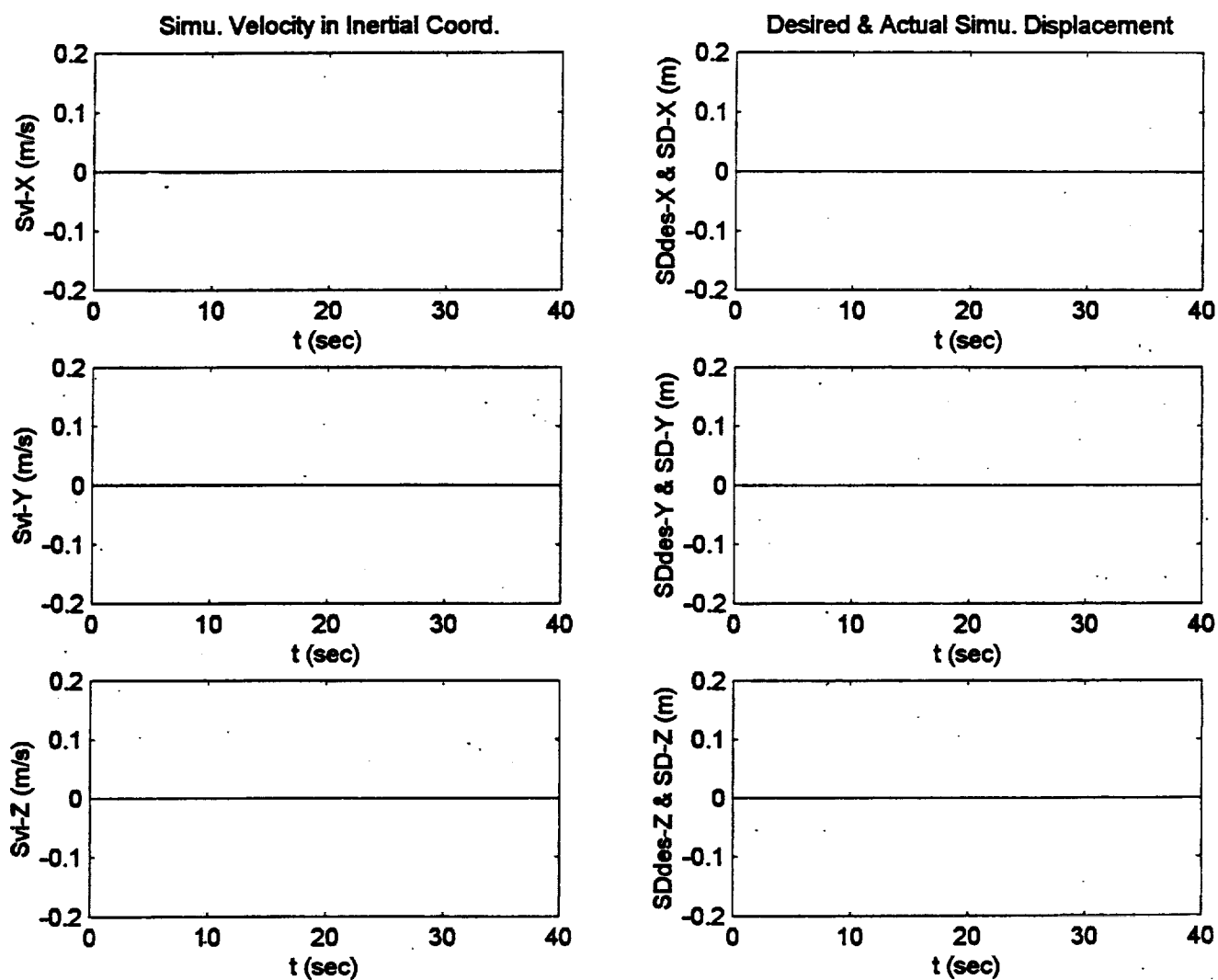


Figure C.23. NASA Adaptive Algorithm Yaw Doublet Pulse Input.

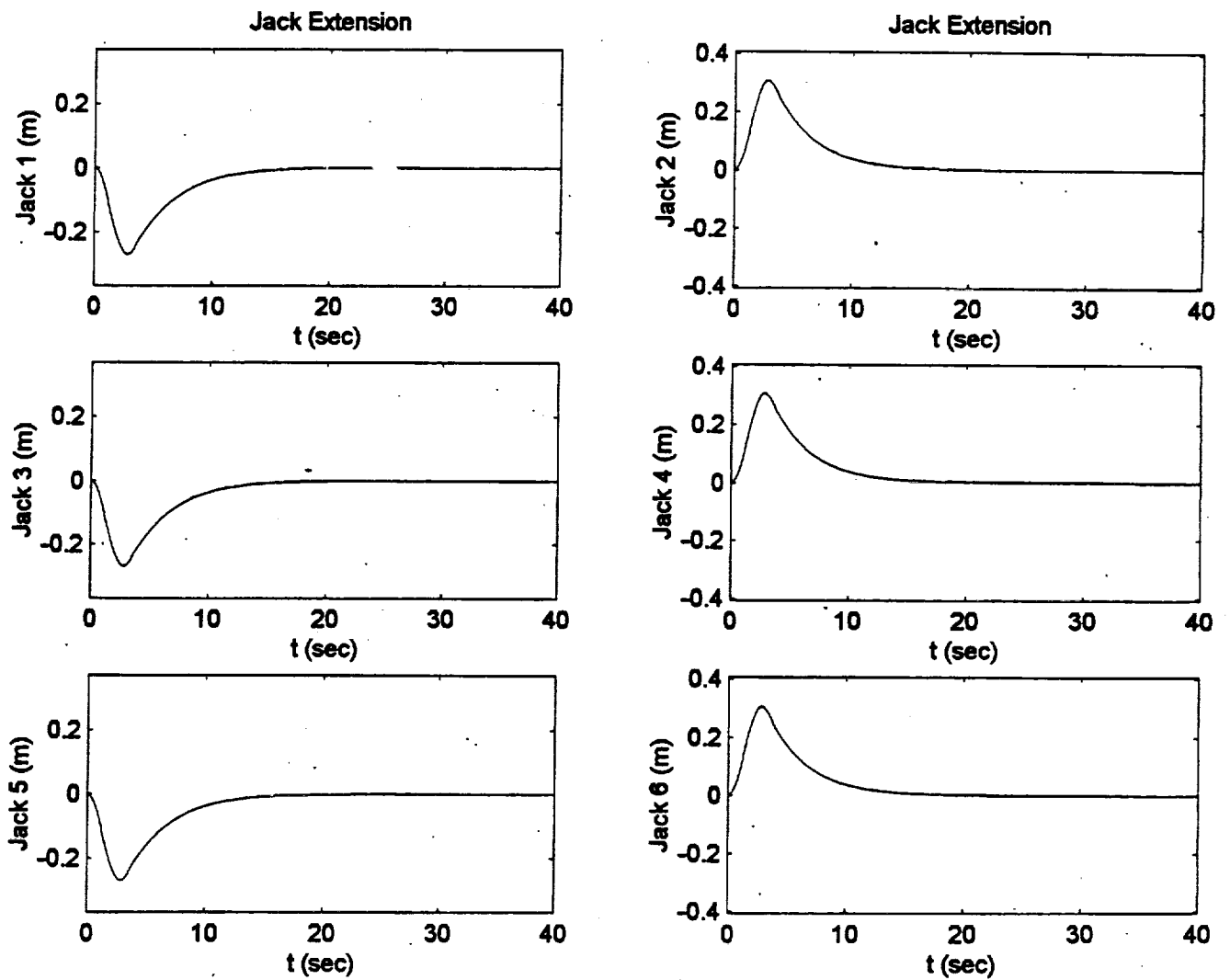


Figure C.23. NASA Adaptive Algorithm Yaw Doublet Pulse Input.

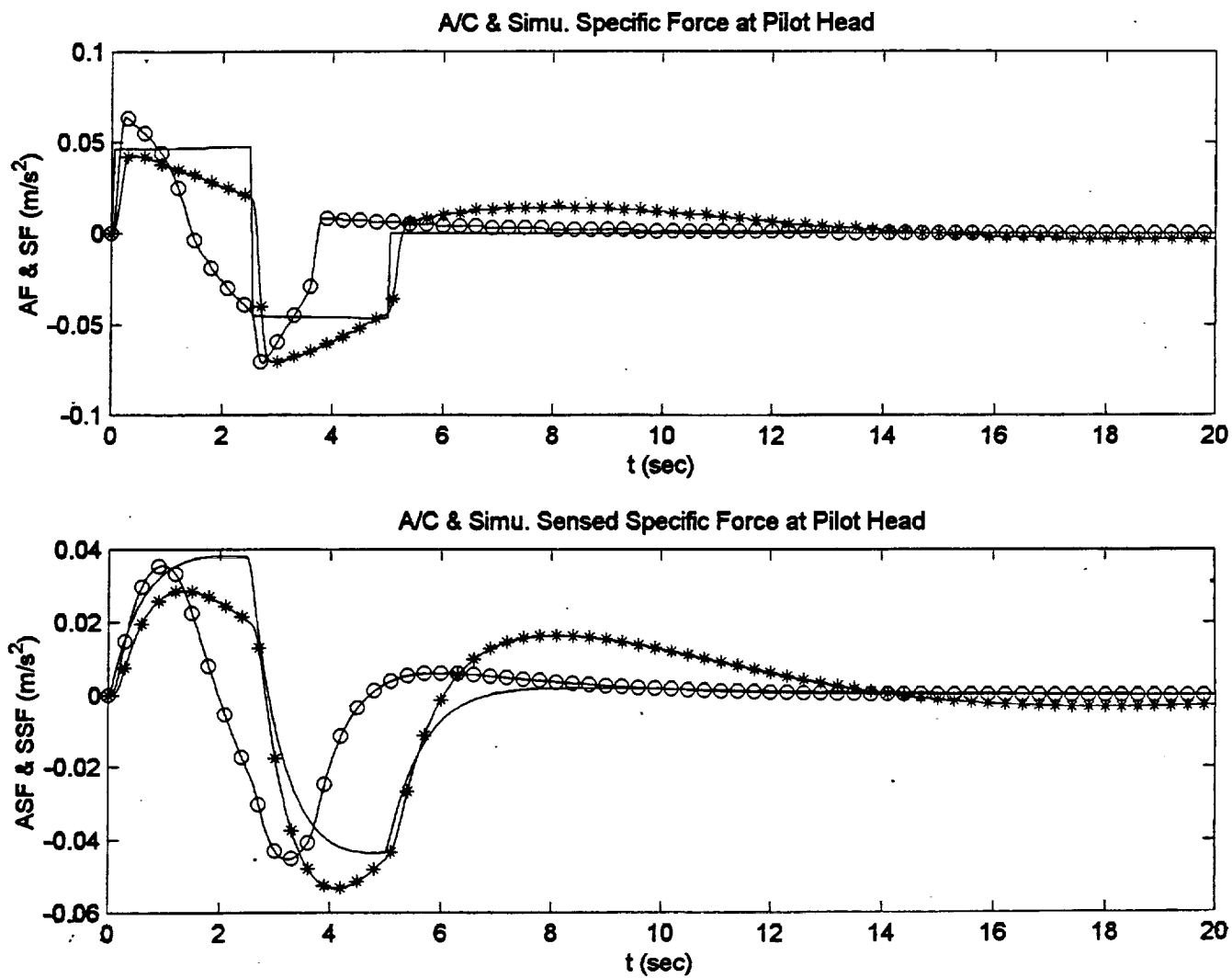


Figure C24. Yaw Doublet Pulse Input Specific Force Comparison.

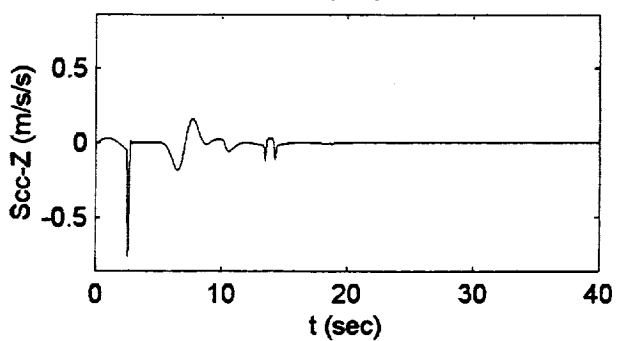
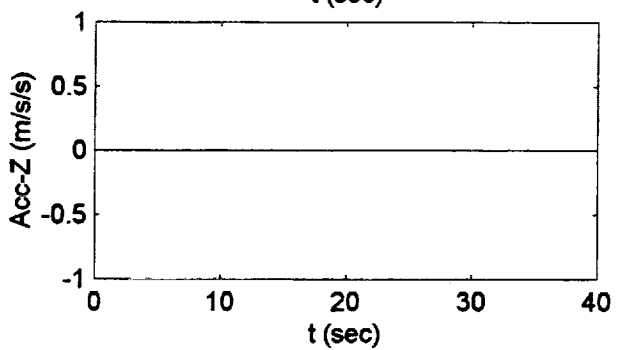
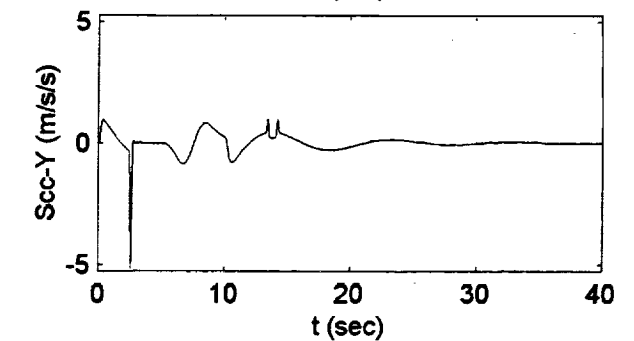
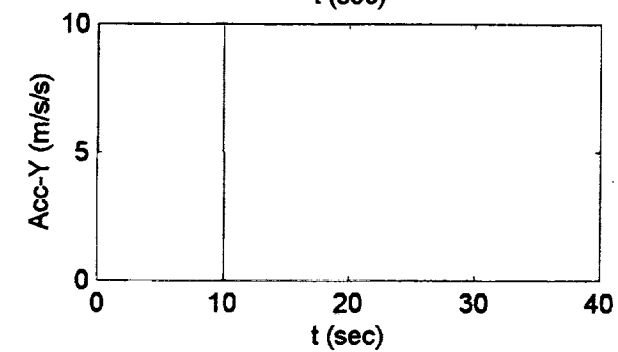
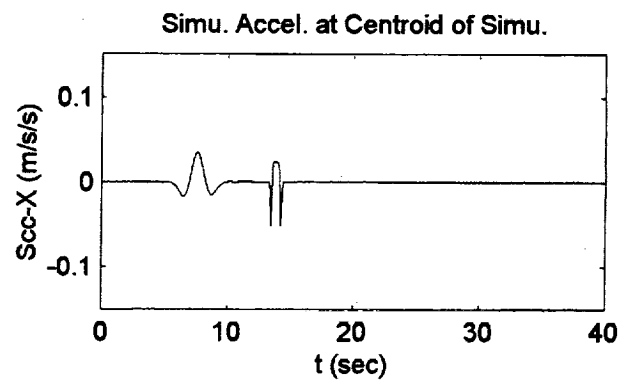
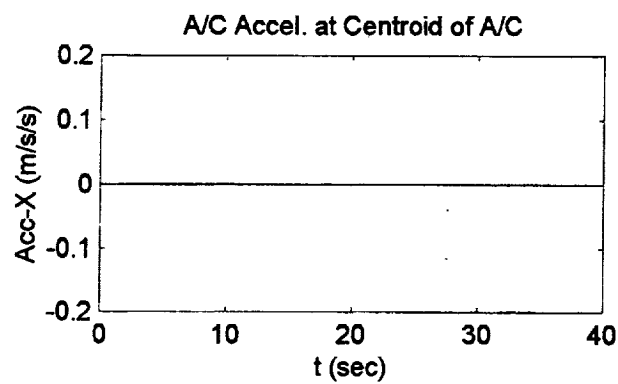


Figure C25. Sway Pulse Input Braking Algorithm Test.

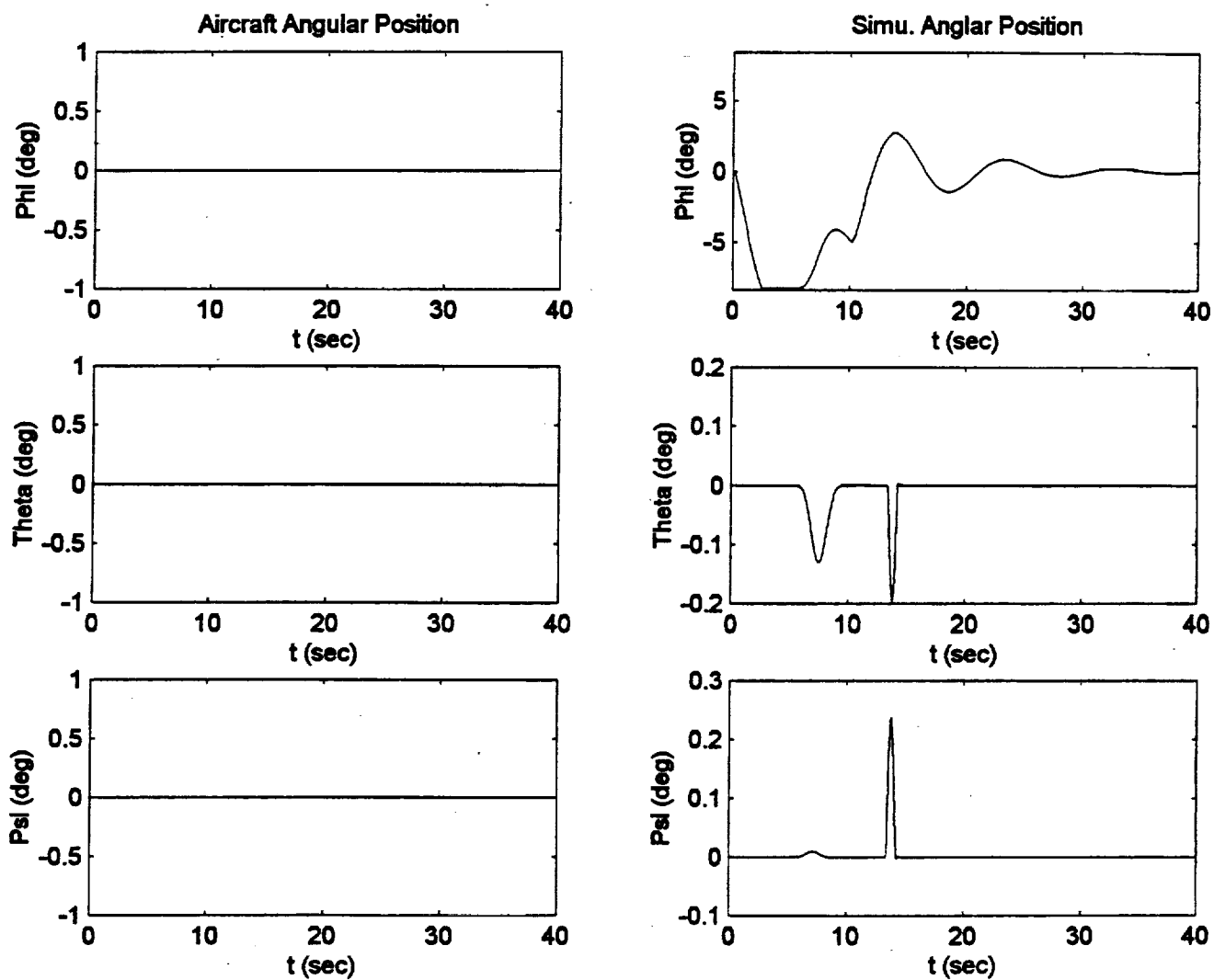


Figure C25. Sway Pulse Input Braking Algorithm Test.

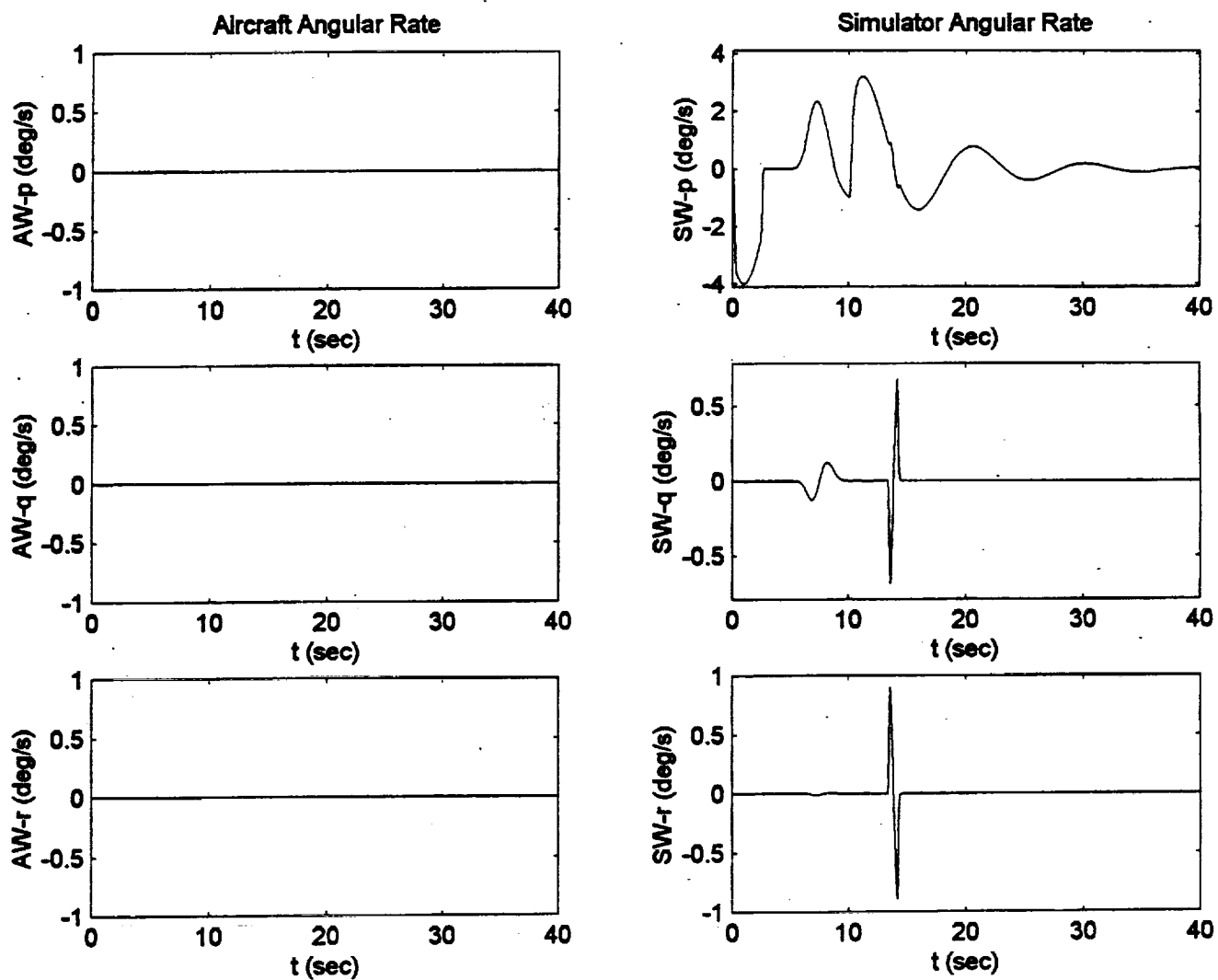


Figure C25. Sway Pulse Input Braking Algorithm Test.

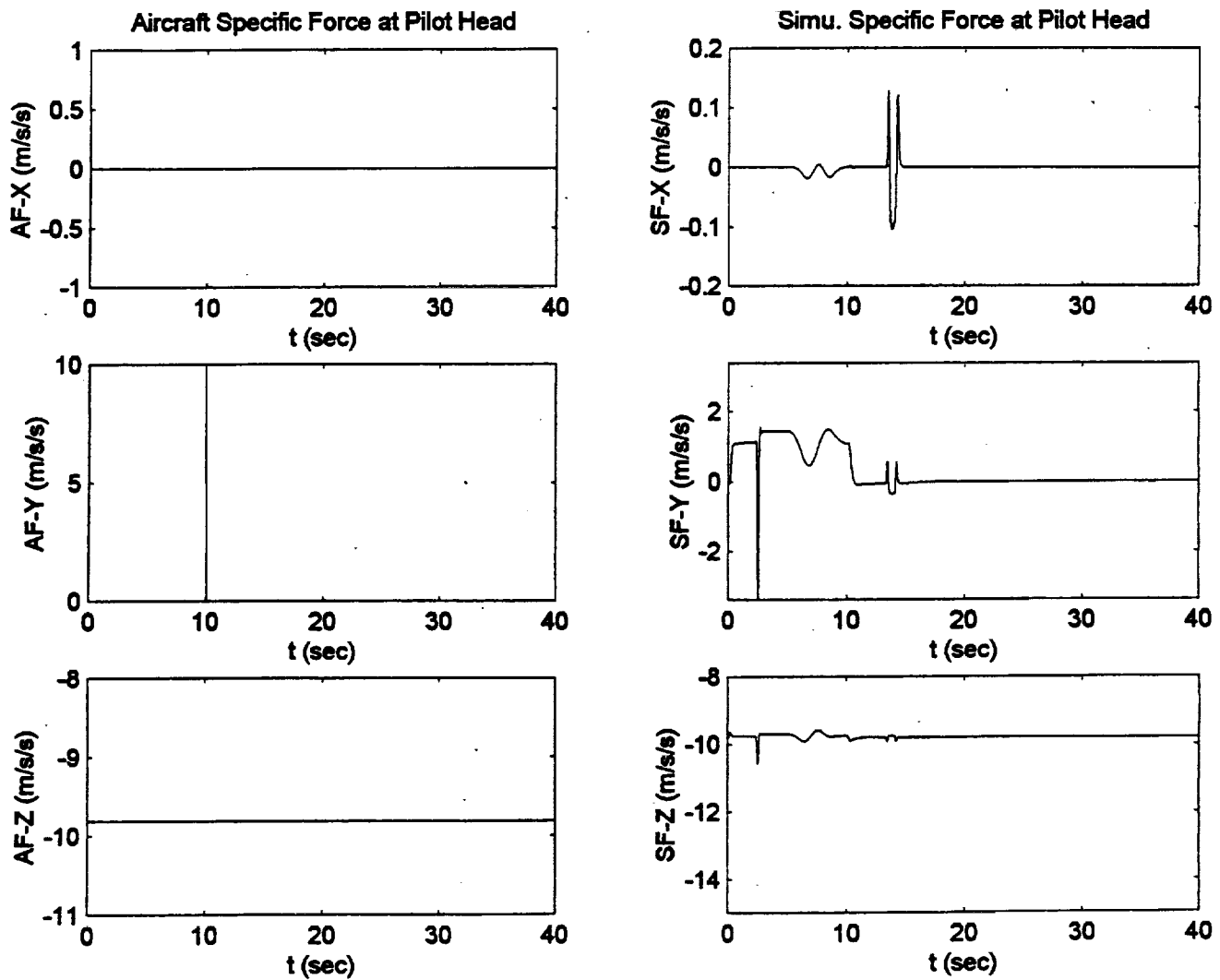


Figure C25. Sway Pulse Input Braking Algorithm Test.

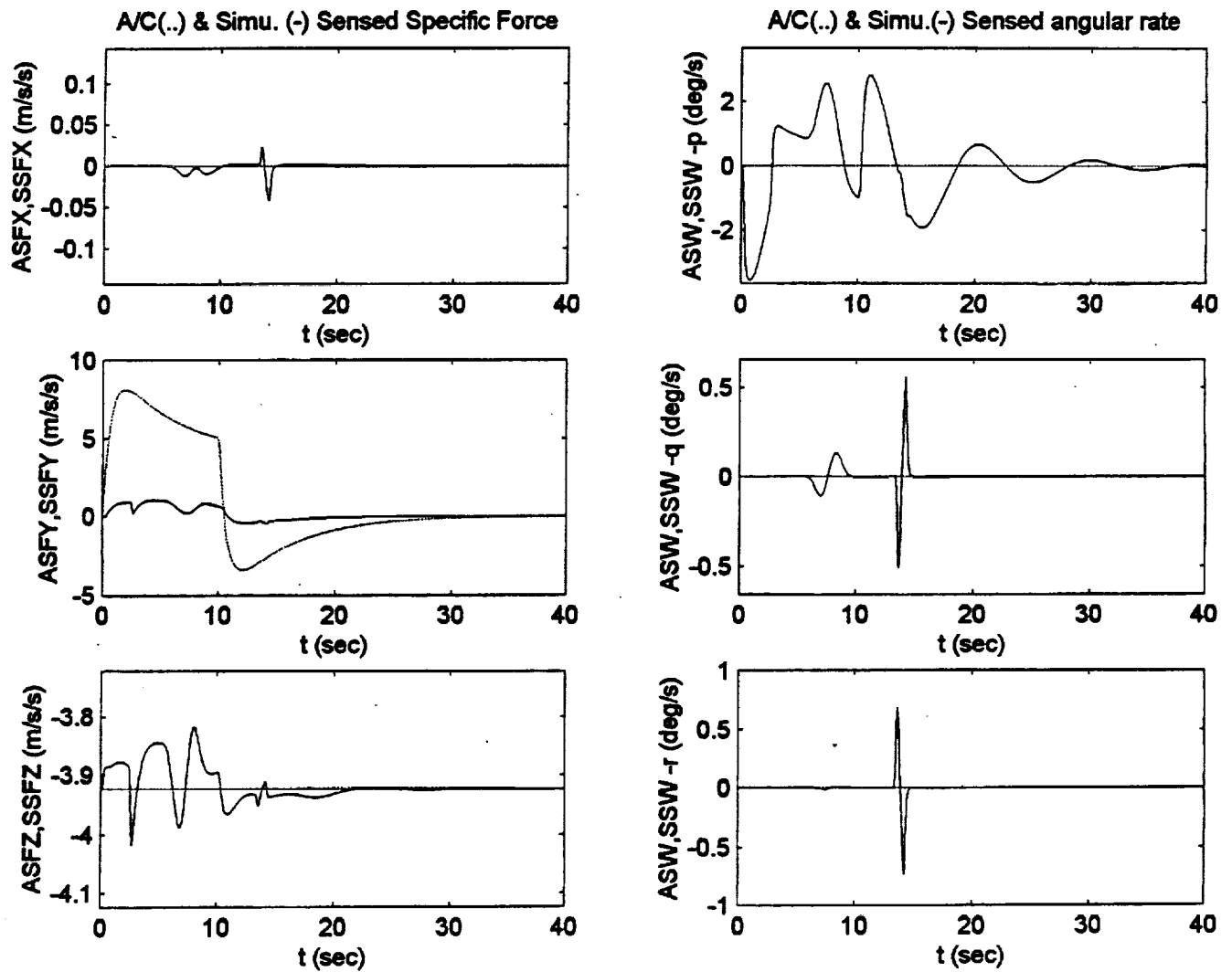


Figure C25. Sway Pulse Input Braking Algorithm Test.

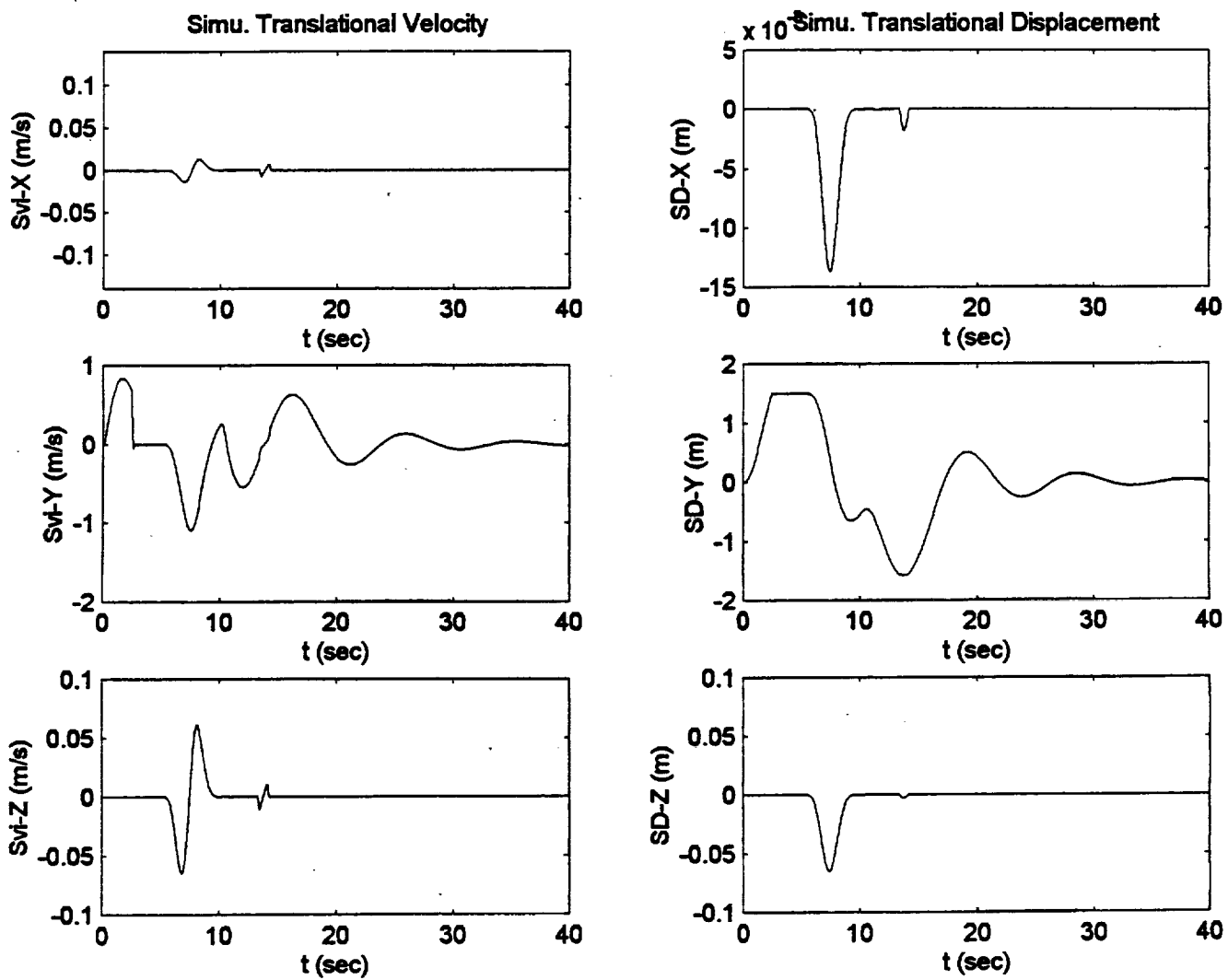


Figure C25. Sway Pulse Input Braking Algorithm Test.

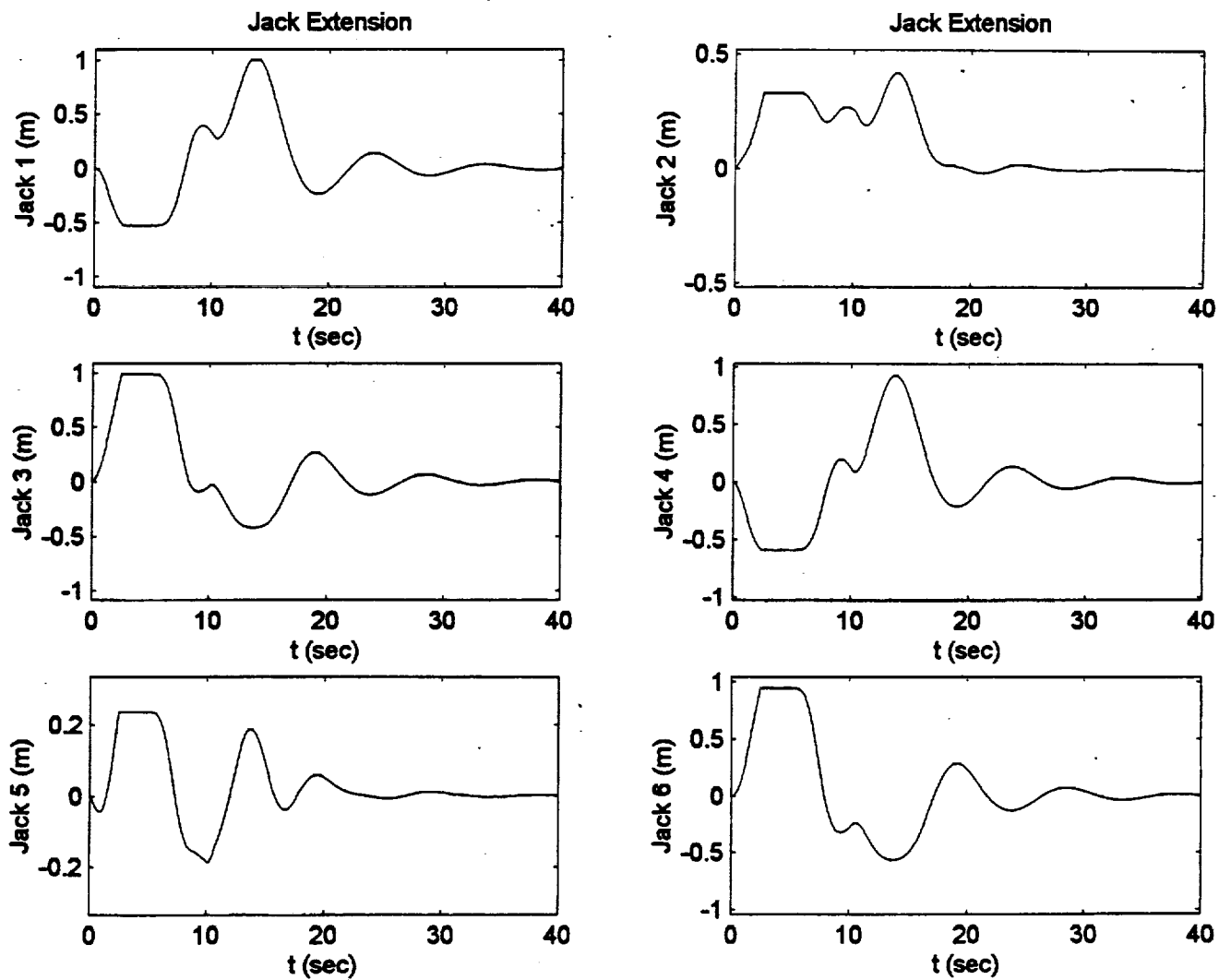


Figure C25. Sway Pulse Input Braking Algorithm Test.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE March 2000		3. REPORT TYPE AND DATES COVERED Contractor Report
4. TITLE AND SUBTITLE Motion Cueing Algorithm Development: Initial Investigation and Redesign of the Algorithms			5. FUNDING NUMBERS 992-30-11-01 NAS1-20454	
6. AUTHOR(S) Robert J. Telban, Weimin Wu, and Frank M. Cardullo				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) State University of New York Binghamton, NY 13902-6000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Langley Research Center Hampton, VA 23681-2199			10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA/CR-2000-209863	
11. SUPPLEMENTARY NOTES Prepared by the State University of New York - Binghamton for NASA Langley Research Center under subcontract to Unisys Corporation, Inc. NASA Langley Research Center Technical Monitor: Jacob A. Houck				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified-Unlimited Subject Category 54 Distribution: Standard Availability: NASA CASI (301) 621-0390			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) In this project four motion cueing algorithms were initially investigated. The classical algorithm generated results with large distortion and delay and low magnitude. The NASA adaptive algorithm proved to be well tuned with satisfactory performance, while the UTIAS adaptive algorithm produced less desirable results. Modifications were made to the adaptive algorithms to reduce the magnitude of undesirable spikes. The optimal algorithm was found to have the potential for improved performance with further redesign. The center of simulator rotation was redefined. More terms were added to the cost function to enable more tuning flexibility. A new design approach using a Fortran/Matlab/Simulink setup was employed. A new semicircular canals model was incorporated in the algorithm. With these changes results show the optimal algorithm has some advantages over the NASA adaptive algorithm. Two general problems observed in the initial investigation required solutions. A nonlinear gain algorithm was developed that scales the aircraft inputs by a third-order polynomial, maximizing the motion cues while remaining within the operational limits of the motion system. A braking algorithm was developed to bring the simulator to a full stop at its motion limit and later release the brake to follow the cueing algorithm output.				
14. SUBJECT TERMS Flight Simulation, Simulators, Motion Systems, Cueing Algorithms Motion Perception			15. NUMBER OF PAGES 287	
			16. PRICE CODE A13	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	